it Series

A Comprehensive Study of

OPERATING SYSTEMS

& NETWORKS

Also Includes:

- More than 450 Short Questions
- More than 300 Multiple Choice
- More than 200 True/False
- Windows and Linux

5th Edition

Tariq Mahmood
Imran Saeed

Reviewed by:
Tasleem Mustafa

IT Series

A Comprehensive Study of

Operating Systems & Networks

5th Edition

Written by

Tariq Mahmood
M.Sc. (CS), MCP, JCP

Imran Saeed
M.Sc. (CS), MCP, JCP

Reviewed by

Tasleem Mustafa
MS (CS), MSc (CS), M.Sc. (Math), MIS (USA), OCP
Chairman, Deptt. of Computer Science,
University of Agriculture, Faisalabad, Pakistan.

Kitab Markaz, Faisalabad

All Rights Reserved with IT Series

No part of this book may be reproduced in any form or any means without the written permission from **IT Series**.

Ch. Ashfaq Shahid
Advocate, High Court

Published by: IT Series

Edition: 5th

Price: **Rs. 240/-**

Can be had from:

Kitab Markaz
Aminpur Bazar, Faisalabad.
Ph: 041-2642707

Imtiaz Book Depot
Urdu Bazar, Lahore.
Ph: 042-7235944

Preface

This book is about the structure and functions of operating systems and networks. Its purpose is to present the nature, characteristics and working of modern operating systems, as clearly and comprehensively as possible. We have tried our best to provide a thorough discussion of the fundamentals of operating system design and architecture.

The book is very helpful for the students who study the course of operating systems in different universities and colleges, in the classes of MS-CS, MCS, BS-CS, BS-IT and BA/B.Sc.

There are thirteen chapters in the book. Each chapter begins with an outline of the chapter and ends with a number of review questions with answers, which will help the students to great extent. The book also contains a large number of figures and tables to explain the concepts of operating system and networks.

The book also includes comprehensive chapters on **Windows 2000 Professional**. The chapters give fundamental and practical knowledge of the subject. It also includes objective type questions with answers. These questions are specially prepared for examination point of view and will help the students to get higher marks.

Readers are welcome to send suggestions for improvements of the book by sending email to us at **info@itseries.edu.pk**. You can also visit our website **www.itseries.com.pk** for any interaction and latest information about the book.

Authors

Contents

1 Introduction to Operating System 1
1.1 Computer System 2
1.2 Operating System 2
    1.2.1 Operating System Objectives 2
    1.2.2 Role of Operating System 3
1.3 Mainframe Systems 3
    1.3.1 Simple Batch Systems 3
    1.3.2 Multi-Programmed Systems 4
    1.3.3 Time Sharing 5
    1.3.4 Time Sharing Systems 5
1.4 Single-Processor Systems 6
1.5 Multiprocessor Systems 6
    1.5.1 Advantages of Multiprocessor Systems 6
    1.5.2 Types of Multiprocessor Systems 7
1.6 Distributed Systems 7
    1.6.1 Advantages of Distributed Systems 7
1.7 Clustered Systems 8
1.8 Real-Time Systems 9
    1.8.1 Types of Real-Time Systems 9
1.9 Hand-Held Systems 9
1.10 Multimedia Systems 10
1.11 Interrupts 10
    1.11.1 Common Functions of Interrupts 10
    1.11.2 Classes of Interrupts 11
1.12 Operating System Modes 11
1.13 I/O Structure 12
1.14 Operating System Components 12
    1.14.1 Process Management 13
    1.14.2 Main Memory Management 13
    1.14.3 Secondary Storage Management 13
    1.14.4 I/O System Management 13
    1.14.5 File Management 14
    1.14.6 Protection System 14
    1.14.7 Networking (Distributed Systems) 14
    1.14.8 Command Interpreter System 14
1.15 Kernel 15
1.16 Operating System Services 16
1.17 System Calls 17
    1.17.1 Main Types of System Calls 17
Review Questions 18

2 Process Management 22
2.1 Process 23
    2.1.1 Process Attributes 23
2.2 Process States 23
2.3 State Transition in Process 24
2.4 Process Control Block 24
2.5 Process Scheduling 25
    2.5.1 Scheduling Queues 25
    2.5.2 Scheduler 25
    2.5.3 Context Switch 27
2.6 Operations on Processes 27

2.6.1 Process Creation ....
2.6.2 Process Termination 27
2.7 Inter-Process Communication (IPC) 28
2.7.1 Independent Processes 28
2.7.2 Cooperating Processes 28
2.7.3 Shared Memory Model 29
2.7.4 Message Passing Model 29
2.7.4.1 Naming 30
2.7.4.2 Synchronization 31
2.7.4.3 Buffering 32
Review Questions 32
........33

3 Threads ....
3.1 Threads **37**
3.1.1 Benefits of Threads 38
3.1.2 Difference between Processes and Threads 38
3.2 Multithreading 38
3.2.1 Advantages of Multithreading 39
3.3 Types of Threads 39
3.3.1 Kernel-Level Threads 40
3.3.2 User-Level Threads 40
3.4 Multithreading Models 40
3.4.1 Many-to-One Model 41
3.4.2 One-to-one Model 41
3.4.3 Man-to-Many Model 41
3.5 Thread Libraries 42
3.6 Operating-System Examples 42
Review Questions 42
........43

4 CPU Scheduling ....
4.1 CPU Scheduling **46**
4.2 Multiprogramming 47
4.3 Scheduling Objectives 47
4.4 CPU I/O Burst Cycle 48
4.5 Working of CPU Scheduler 48
4.6 Preemptive and Non-Preemptive Scheduling 48
4.7 Interval Timer 49
4.8 Dispatcher 49
4.9 Scheduling Criteria 49
4.10 Scheduling Algorithms 50
4.10.1 First Come First Served Scheduling 50
4.10.2 Shortest Job First Scheduling 51
4.10.2.1 Non-Preemptive SJF Example 52
4.10.2.2 Preemptive SJF example 53
4.10.3 Priority Scheduling 53
4.10.3.1 Non-Preemptive Example 54
4.10.3.2 Preemptive Example 54
4.10.4 Round Robin Scheduling 55
4.10.5 Multi-Level Queue Scheduling 56
4.10.6 Multi-Level Feedback Queue Scheduling 56
4.11 Multiple Processor Scheduling 57
4.12 Real-Time Scheduling 57
4.13 Algorithm Evaluation 58
4.13.1 Deterministic Modeling 58
.... 58

4.13.2 Queuing Models 58
4.13.3 Simulation 58
4.13.4 Implementation 58
Review Questions 59

---

5 Process Synchronization 65
5.1 Process Synchronization 66
5.2 Race Conditions 66
5.3 Critical-Section Problem 67
5.3.1 Criterion to Solve Critical-Sections Problem 67
5.3.2 Two-Process Solution for Critical Section 67
5.3.2.1 Algorithm 1 67
5.3.2.2 Algorithm 2 68
5.3.2.3 Algorithm 3 69
5.3.3 Bakery Algorithm Multiple-Process Solution 70
5.4 Hardware Solution to Synchronization (Test & Set) 71
5.5 Semaphores 72
5.6 Classic Problems of Synchronization 73
5.6.1 Bounded-Buffer Producer-Consumer Problem 73
5.6.2 Readers-Writers Problem 74
5.6.3 The Dining Philosophers Problem 75
5.6.4 Sleeping Barber Problem 76
5.7 Counting Semaphores 77
5.8 Monitors 77
5.8.1 Advantages of Monitors 78
5.8.2 Condition Variable 78
5.8.3 Implementation Questions for Monitors 78
5.8.4 Readers and Writers Problem using Monitors 79
Review Questions 80

---

6 Deadlocks 86
6.1 Deadlock 87
6.2 Deadlock Examples 87
6.3 Resources 88
6.3.1 Types of Resources 88
6.4 Deadlock Characterization 89
6.5 Resource-Allocation Graph 89
6.6 Methods for Handling Deadlocks 90
6.7 Deadlock Prevention 91
6.7.1 Mutual Exclusion 91
6.7.2 Hold and Wait 91
6.7.3 No Preemption 91
6.7.4 Circular Wait 92
6.8 Deadlock Avoidance 92
6.8.1 Safe State 92
6.8.1.1 Safe Sequence 93
6.8.1.2 Unsafe Sequence 93
6.8.2 Resource-Allocation Graph Algorithm 93
6.8.3 Banker's Algorithm 94
6.9 Deadlock Detection 97
6.9.1 Single Instance of Each Resource Type 97
6.9.2 Detection-Algorithm Usage 97
6.10 Deadlock Recovery 97
6.10.1 Process Termination 97

6.10.2 Resource Preemption 98

Review Questions 99

---

7 Memory Management 101

7.1 Memory Management 102

7.2 Basic Requirements 102

- 7.2.1 Relocation 102
- 7.2.2 Protection 102
- 7.2.3 Sharing 103
- 7.2.4 Program Translation and Address Spaces 103
- 7.2.5 Loading and Linking 103
  - 7.2.5.1 Absolute Loading 103
  - 7.2.5.2 Re-Locatable Loading 104
  - 7.2.5.3 Dynamic Loading (Dynamic Relocation) 104
  - 7.2.5.4 Linking 104
  - 7.2.5.5 Overlaying 104

7.3 Contiguous & Non-Contiguous Storage Allocation 105

7.4 Swapping 105

7.5 Memory Allocation Methods 105

- 7.5.1 Single Process Allocation Scheme/Single Absolute Partition 106
- 7.5.2 Multiple Fixed Partition Allocation Scheme 106
- 7.5.3 Multiple Variable Partition Allocation Scheme 107
  - 7.5.3.1 Coalescing Holes 108
  - 7.5.3.2 Storage Replacement Strategies 109
  - 7.5.3.3 Compaction 109

7.6 Logical vs. Physical Address Space 110

7.7 Paging 110

7.8 Page Table Structure and Hardware Support 113

7.9 Segmentation 115

- 7.9.1 Difference between Paging and Segmentation 117

7.10 Segmentation with Paging 118

Review Questions 119

---

8 Virtual Memory 124

8.1 Virtual Memory 125

8.2 Demand Paging 125

8.3 Page Faults / Segment Faults 126

8.4 Page Replacement 126

8.5 Page Replacement Algorithm 127

- 8.5.1 First-In First-Out (FIFO) 127
- 8.5.2 Page Optimal Replacement 128
- 8.5.3 Least Recently Used (LRU) 128
- 8.5.4 Not Recently Used (NRU) 128
- 8.5.5 LRU Approximation Algorithms 129
  - 8.5.5.1 Additional-Reference Bits Algorithm 129
  - 8.5.5.2 Second-Chance Algorithm 129
  - 8.5.5.3 Enhanced Second-Chance Algorithm 129

8.6 Allocation of Frames 129

- 8.6.1 Minimum Number of Frames 130
- 8.6.2 Algorithms 130
- 8.6.3 Replacement Scope 130

8.7 Thrashing 131

8.8 Working Set 132

8.9 Page Size 132

Review Questions 133

9 File System Management 136

9.1 File 137
    9.1.1 File Attributes 137
    9.1.2 File Operations 137
9.2 Open Files 138
9.3 Types of File System Objects 138
9.4 Information Types 138
9.5 File Management Systems 139
9.6 File System Architecture 139
    9.6.1 File Management Functions 140
9.7 Access Methods 141
    9.7.1 Sequential Access 141
    9.7.2 Direct Access 141
    9.7.3 Other Access Methods 142
9.8 File Locking 142
9.9 File Blocking 142
9.10 File Protection 143
    9.10.1 File Naming 143
    9.10.2 Passwords 143
    9.10.3 Access Control 144
9.11 File System Reliability 144
    9.11.1 Periodic Dump 144
    9.11.2 Incremental Dump 144
9.12 File Management Module 145
    9.12.1 Accessing and Organizing 145
    9.12.2 Management of Secondary Memory Space 146
        9.12.2.1 Contiguous Blocks 147
        9.12.2.2 Block Linkage 147
        9.12.2.3 Index Blocks 148
9.13 Directories and Names 149
9.14 Pre-Process Root Directory 149
9.15 Directory Structure 149
    9.15.1 Operations Performed on Directory 150
    9.15.2 Single Level Directory 150
    9.15.3 Two-Level Directory 151
    9.15.4 Tree-Structured Directory 151
    9.15.5 Acyclic-Graph Directories 152
    9.15.6 General Graph Directories 152
9.16 Directory Entries 153

Review Questions 153

10 I/O & Device Management 157

10.1 I/O Management Module 158
10.2 Objectives 159
10.3 Buffering 160
10.4 Spooling 160
10.5 Handshaking 161
10.6 Interrupts 161
    10.6.1 Software Poll 162
    10.6.2 Vectored Interrupts 162
10.7 Interrupt Priorities 163
    10.7.1 Disable interrupts 163
10.8 Classes of Interrupts 164
10.9 I/O Devices 165
10.10 Storage hierarchy 165

10.11 Hardware Organization 166
    10.11.1 Single-Bus Architecture 166
    10.11.2 Hardware Organization 167
    10.11.3 I/O Control 167
    10.11.4 Ports and Memory-Mapped I/O 167
    10.11.5 Module Registers 168
    10.11.6 Busy Wait I/O 168
    10.11.7 Polled I/O 168
    10.11.8 Interrupt I/O 168
    10.11.9 Direct Memory Access 168
10.12 Software Organization 169
    10.12.1 Network I/O 169
    10.12.2 Logical I/O 169
    10.11.3 Buffering 170
    10.12.4 Caching 170
    10.12.5 Device Drivers 170
Review Questions 171

---

11 Operating System Security 172
11.1 Security 173
11.2 Security Problem 173
11.3 Authentication 173
    11.3.1 Passwords 173
    11.3.2 Password Vulnerabilities 174
    11.3.3 Encrypted Passwords 174
    11.3.4 One Time Passwords 174
11.4 Prevention 174
11.5 Detection 175
11.6 Correction 175
11.7 Identification 175
11.8 Program Threats 175
    11.8.1 Trojan Horse 176
    11.8.2 Trap Doors 176
11.9 System Threats 176
    11.9.1 Worms 176
    11.9.2 Viruses 176
11.10 Threat Monitoring 176
11.11 System Security 177
Review Questions 178

---

12 Case Study: Linux 179
12.1 Linux 180
12.2 Brief History of Linux 180
12.3 Features of Linux 180
12.4 Who is using Linux? 181
12.5 Linux Distributions 182
12.6 Linux Architecture: System Structure 182
12.7 Process Scheduler Subsystem 183
    12.7.1 Goals 183
    12.7.2 External Interface 183
    12.7.3 Subsystem Description 183
    12.7.4 Data Structures 184
    12.7.5 Subsystem Structure 184

12.7.6 Subsystem Dependencies 185
12.8 Memory Manager Subsystem 185
12.8.1 Goals 185
12.8.2 External Interface 186
12.8.3 Subsystem Description 186
12.8.4 Data Structures 187
12.8.5 Subsystem Structure 187
12.8.6 Subsystem Dependencies 188
12.9 Virtual File System 189
12.9.1 Goals 189
12.9.2 External Interface 189
12.9.3 Subsystem Description 190
12.9.4 Data Structures 191
12.9.5 Subsystem Structure 192
12.9.6 Subsystem Dependencies 192
12.10 Inter-Process Communication 193
12.10.1 Goals 193
12.10.2 External Interface 193
12.10.3 Subsystem Description 194
12.10.4 Data Structures 195
12.10.5 Subsystem Structure 196
12.10.6 Subsystem Dependencies 197
12.11 Network Interface 197
12.11.1 Goals 197
12.11.2 External Interface 197
12.11.3 Subsystem Description 198
12.11.4 Data Structures 199
12.11.5 Subsystem Structure 199
12.11.6 Subsystem Dependencies 200
Review Questions 201

Operating Systems: Objective Type Questions with Answers 202

13 Data Communications and Networks 238
13.1 Data Communication 241
13.1.1 Characteristics of Data Communication System 241
13.1.2 Basic Elements of Data Communication 241
13.2 Data Transmission Modes 242
13.3 Analog and Digital Data 243
13.4 Forms of Data Transmission 243
13.4.1 Digital Data Transmission 243
13.4.2 Analog Data Transmission 244
13.5 Forms of Signals 244
13.6 Modulation 245
13.6.1 Types of Modulation 245
13.7 Serial and Parallel Transmission 246
13.8 Types of Data Transmission 247
13.8.1 Asynchronous Transmission 247
13.8.2 Synchronous Transmission 247
13.9 Computer Network 248
13.9.1 Examples of Computer Network 248
13.9.2 Network Criteria 248
13.9.3 Type of Connection 249

13.9.4 Benefits / Advantages of Computer Networks 250
13.9.5 Disadvantages of Networks 250
13.10 Client and Server 251
13.10.1 Dedicated Server 251
13.10.2 Authentication Server 252
13.10.3 File Server 252
13.11 Types of Computer Networks 252
13.11.1 LAN (Local Area Network) 254
13.11.1.1 Peer-to-Peer Network 255
13.11.1.2 Client-Server Network 255
13.11.2 WAN (Wide Area Network) 256
13.11.2.1 Internet Peer-to-Peer 257
13.11.3 MAN (Metropolitan-Area Network) 258
13.11.4 Difference between LAN and WAN 259
13.12 Wireless Networks 259
13.12.1 Categories of Wireless Networks 259
13.13 Network Topologies 260
13.13.1 Bus Topology 260
13.13.2 Star Topology 260
13.13.3 Ring Topology 261
13.13.4 Tree Topology 261
13.13.5 Mesh Topology 262
13.14 Network Standards 262
13.14.1 Standards Organization 262
13.15 Protocol 263
13.15.1 Functions of Protocols 263
13.15.2 Key Elements of Protocol 263
13.16 Network Communication Technologies 264
13.16.1 Ethernet 264
13.16.1.1 CSMA/CD 264
13.16.2 Token Ring 265
13.16.3 TCP/IP 265
13.16.4 Bluetooth 265
13.16.5 Wi-Fi 265
13.16.6 UWB 267
13.16.7 IrDA 267
13.16.8 WiMAX 267
13.16.9 WAP 267
13.17 Home Networks 267
13.18 Communications Software 268
13.19 Switched Networks 268
13.19.1 Circuit Switching 268
13.19.2 Packet Switching 268
13.19.3 Message Switching 269
13.20 Telephone Network 270
13.20.1 Dial-Up Lines 270
13.20.2 Dedicated Lines 270
13.21 Communication Devices 271
13.21.1 Modem 272
13.21.2 Network Interface Card 274
13.21.3 Network Bridge 275
13.21.4 Wireless Access Point 276
13.21.5 Hub 276
13.21.6 Network Switch 277
13.21.7 Managed Switches 277
13.21.8 Multilayer Switches 277
13.21.9 Routers 278

13.21.10 Gateway 279
13.22 Internetworks 279
13.23 Communications Channel 279
13.23.1 Physical Transmission Media 280
13.23.1.1 Twisted Pair 280
13.23.1.2 Coaxial Cable 281
13.23.1.3 Fiber Optics 281
13.23.2 Wireless Transmission Media 282
13.23.2.1 Microwave System 282
13.23.2.2 Satellite Communication 283
13.23.2.3 Broadcast Radio 283
13.23.2.4 Cellular Communication 283
13.23.2.5 Infrared 283
13.24 Media Connectors 285
13.25 Transmission Impairment 288
13.25.1 Causes of Transmission Impairment 288
13.26 Fiber Distributed Data Interface (FDDI) 289
13.27 OSI Model 291
13.27.1 Advantages of OSI Model 291
13.27.2 Layered Architecture 291
13.27.3 Peer-to-Peer Processes 292
13.27.4 Interfaces between Layers 292
13.27.5 Organization of the Layers 293
13.27.6 Layers of OSI Model 293
13.28 TCP / IP Protocol Suite 296
13.28.1 Physical and Data Link Layers 297
13.28.2 Network Layer 297
13.28.3 Transport Layer 298
13.28.4 Application Layer 298
13.29 TCP/UDP Port Functions 298
13.30 IP Addressing 299
13.30.1 IPv4 Addressing 299
13.30.2 IPv6 Addressing 302
13.30.3 IP Address Assignment 303
13.31 Protocols 303
13.31.1 TCP 304
13.31.2 UDP 305
13.31.3 DHCP 305
13.31.4 SMTP 306
13.31.5 SNMP 306
13.31.6 File Transfer Protocol (FTP) 306
13.31.7 Secure File Transfer Protocol (SFTP) 307
13.31.8 Trivial File Transfer Protocol (TFTP) 307
13.31.9 POP3 and IMAP4 307
13.31.10 Hypertext Transfer Protocol (HTTP) 307
13.31.11 Hypertext Transfer Protocol Secure (HTTPS) 307
13.32 NAT 307
13.33 ICS 308
13.34 DNS 308
13.35 Static and Dynamic IP Addresses 308
13.36 Firewalls 309
Review Questions 309
Multiple Choices 323
Fill in the Blanks 329
True/False 332

14 Windows 2000 Professional: Installation and Configuration 334
14.1 Windows 2000 Professional 336
    14.1.1 Features of Windows 2000 Professional 336
14.2 Workgroup and Domain 337
    14.2.1 Workgroup 337
    14.2.2 Domain 338
14.3 Disk Partition 338
    14.3.1 Types of Partitions 338
    14.3.2 Volumes 339
14.4 File Systems 339
    14.4.1 FAT 340
    14.4.2 FAT32 340
    14.4.3 NTFS 341
    14.4.4 CDFS 341
    14.4.5 Difference between FAT and NTFS 342
    14.4.6 Disk Management Utility 342
14.5 Windows Operating System 342
    14.5.1 Peer-to-peer Workgroup 342
    14.5.2 Server Centric Network 342
    14.5.3 Installing Windows 2000 from CD-ROM 343
    14.5.4 Installing Windows 2000 over a Network 350
14.6 Windows 2000 Desktop 351
    14.6.1 Elements of Desktop 352
    14.6.2 Desktop Properties 352
    14.6.3 Wallpaper 353
    14.6.4 Screen Saver 353
        14.6.4.1 Energy Saver Feature 354
    14.6.5 Appearance 354
    14.6.6 Effects 354
    14.6.7 Display Properties 355
    14.6.8 Date and Time 355
    14.6.9 Mouse Properties 355
    14.6.10 Keyboard Properties 356
14.7 Shortcut 357
    14.7.1 Creating Shortcuts 357
14.8 Accessibility Features 358
    14.8.1 Accessibility Wizard 358
14.9 Control Panel 359
    14.9.1 Elements of Control Panel 361
14.10 Computer Management 361
    14.10.1 Using Microsoft Management Console 361
    14.10.2 Using Administrative Tools 361
14.11 Using Printers with Windows 361
    14.11.1 Installation of Local Printer 361
    14.11.2 Connecting to a Network Printer 362
    14.11.3 Managing Printer Queue 363
14.12 Adding and Removing Hardware 363
    14.12.1 Adding Null Modem 363
    14.12.2 Installing a Network Adapter 367
    14.12.3 Configuring Network Adaptor 368

15 Managing Users, Files & Folders 371
15.1 User Accounts 372
    15.1.1 Types of User Accounts 372
    15.1.2 Creating User Account 372
15.2 Groups 375
    15.2.1 Local Groups 375
    15.2.2 Built-in Groups 377
15.3 Auditing 377
    15.3.1 Audit Policy 378
    15.3.2 Planning an Audit Policy 378
    15.3.3 Implementing Audit Policy 379
    15.3.4 Setting an Audit Policy 379
15.4 Account Policies 380
    15.4.1 Password Policy 380
    15.4.2 Account Lockout Policy 381
15.5 Security Options 381
15.6 Group Policy Object and Active Directory 381
    15.6.1 Active Directory 385
15.7 Profiles 385
    15.7.1 Types of Profiles 386
    15.7.2 Logon Script 386
    15.7.3 Setting up Home Folder 386
15.8 File and Folder Management 386
    15.8.1 Organizing Files and Folders 387
    15.8.2 Managing Folder Options 388
    15.8.3 Searching for Files and Folders 390
15.9 Shared Folders 391
    15.9.1 Creating a Shared Folder 392
    15.9.2 Shared Folder Permissions 392

16 Network Protocols, Connectivity & Backup 393
16.1 Network Protocols 394
16.2 Dial-up Networking and Internet Connectivity 398
    16.2.1 Creating Dialup Connection 400
16.3 Windows Backup 402
    16.3.1 Types of Backup 403
    16.3.2 Backing up Files and Folders 403

Windows 2000 Professional: Review Questions 405

1

INTRODUCTION TO OPERATING SYSTEM

Chapter Overview

- 1.1 Computer System
- 1.2 Operating System
    - 1.2.1 Operating System Objectives
    - 1.2.2 Role of Operating System
- 1.3 Mainframe Systems
    - 1.3.1 Simple Batch Systems
    - 1.3.2 Multi-Programmed Systems
    - 1.3.3 Time Sharing
    - 1.3.4 Time Sharing Systems
- 1.4 Single-Processor Systems
- 1.5 Multiprocessor Systems
    - 1.5.1 Advantages of Multiprocessor Systems
    - 1.5.2 Types of Multiprocessor Systems
- 1.6 Distributed Systems
    - 1.6.1 Advantages of Distributed Systems
- 1.7 Clustered Systems
- 1.8 Real-Time Systems
    - 1.8.1 Types of Real-Time Systems
- 1.9 Hand-Held Systems
- 1.10 Multimedia Systems
- 1.11 Interrupts
    - 1.11.1 Common Functions of Interrupts
    - 1.11.2 Classes of Interrupts
- 1.12 Operating System Modes
- 1.13 I/O Structure
- 1.14 Operating System Components
    - 1.14.1 Process Management
    - 1.14.2 Main Memory Management
    - 1.14.3 Secondary Storage Management
    - 1.14.4 I/O System Management
    - 1.14.5 File Management
    - 1.14.6 Protection System
    - 1.14.7 Networking (Distributed Systems)
    - 1.14.8 Command Interpreter System
- 1.15 Kernel
- 1.16 Operating System Services
- 1.17 System Calls
    - 1.17.1 Main Types of System Calls

Review Questions

1.1 Computer System

The computer system can be divided into four components as follows:

1. Hardware	Hardware provides basic computing resources. It includes memory, CPU, storage devices, I/O devices etc.
2. Operating System	Operating system controls and coordinates the use of hardware among various application programs for users.
3. Application Programs	Application programs define the ways in which system resources are used to solve computing problems. Application programs include word processing, spreadsheet and database systems etc.
4. Users	The users include people, machines and other computers.

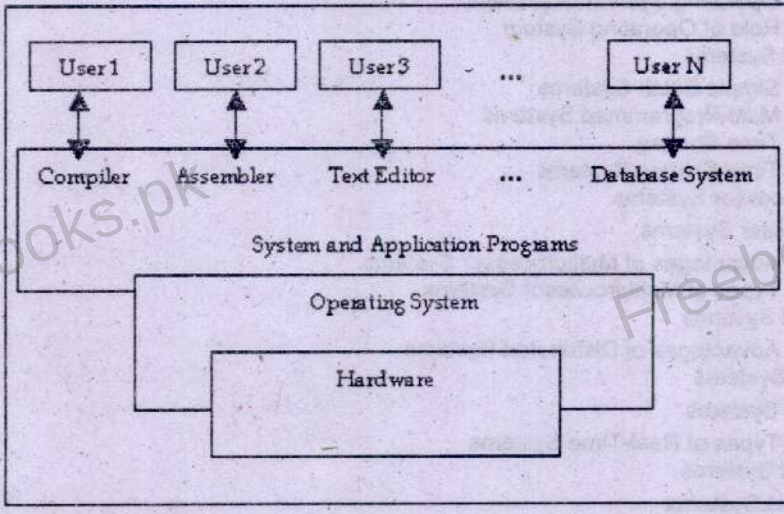


Figure: Conceptual view of a computer system

1.2 Operating System

An **operating system** is a program that manages computer hardware. It is an important part of a computer system. It provides a basis for application programs and acts as an interface between the computer user and the hardware. The operating system controls and coordinates the use of hardware among various application programs for various users.

1.2.1 Operating System Objectives

Different objectives of operating system are as follows:

1. Convenience

The primary objective of an operating system is **convenience** for user. Operating systems execute user programs and make it easy to solve user problems. They also make computer system convenient to use.

2. Efficiency

A secondary objective is the **efficient** operation of computer system. It is particularly important for large, shared multi-user systems. Operating systems can solve this goal. An operating system provides the facility to use the computer hardware in an efficient manner.

3. Ability to evolve

Operating system allows effective development, testing and introduction of new system functions without interfering with the service.

1.2.2 Role of Operating System

The hardware provides the basic computing resources in a computer system. The application programs define the way in which these resources are used to solve computing problems of the users. The operating system controls the hardware and coordinates its use among various application programs for different users.

The role of operating system can be defined in the following ways:

1. Government

Operating system works as a government. The basic resources of computer system are provided by its hardware, software and data. Operating system provides the means and the environment for proper use of its resources.

2. Resource Allocator

A computer system has many resources. Each resource can be used to solve a problem. The operating system is responsible for managing the use of system resources. There may be many requests for a particular resource. These requests may be conflicting. The operating system decides which request should obtain the required resource. The operating system should take into consideration both system efficiency and fairness to individual requests.

3. Control Program

Operating system controls the execution of user programs and devices. It is responsible to monitor and control the operation of software. It controls the hardware components of a computer system. It also ensures that system resources are used properly.

1.3 Mainframe Systems

Mainframe systems were used to solve commercial and scientific problems. Following is the brief description of this type of systems.

1.3.1 Simple Batch Systems

Early computers were run from console. The card readers and tape drives were input devices. Line printers, tape drives and card punched were common output devices. The user did not interact directly with computer system. The user prepared a job that consisted of program, data and some control information (control cards). He then submitted it to the computer operator in the form of punched cards.

The operating system was simple. Its basic job was to transfer control automatically from one job to the next. The operating system was always resident in memory. The operator batched similar jobs together and then ran in the computer to speed up the processing.

The CPU is often idle in this environment as speed of I/O devices is much slower than CPU. After sometime, the introduction of disk instead of card reader resulted in faster I/O devices. In disk technology, the operating system keeps all jobs on a disk instead of card reader. The resources are utilized and jobs are performed more efficiently with the help of job scheduling. **Job scheduling** is possible because all jobs are present on the disk.

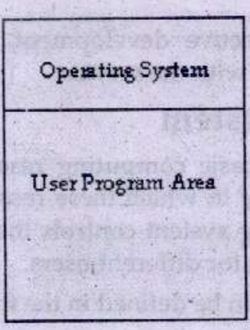


Figure: Memory Layout for simple batch system

The above figure shows the memory layout of a simple batch system. The memory contains operating system in one part and user program in other part. The user space can contain only one process at a time in batch system. Digital Equipment's **VMS** is an example of a batch operating system.

1.3.2 Multi-Programmed Systems

A group of jobs that are ready to be executed is called **job pool**. Since there is more than one job that can be executed, it is possible for the operating system to make a decision about which job to execute next. That decision keeps CPU utilization as high as possible.

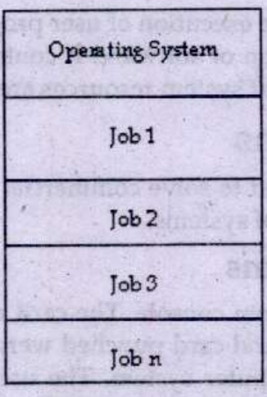


Figure: Memory layout for Multi-programming

In general, it is not possible for a single user to keep CPU or I/O devices busy at all times. Multiprogramming allows the system to increase CPU utilization by ensuring that the CPU always has a job to execute.

The CPU has a pool of jobs. When the currently executing job has to wait (if it is performing some I/O), it is removed from the CPU. Another job is selected and the CPU now executes it. This process ensures that CPU is always executing a job if there is a job to execute. In a non-multiprogrammed system, if a job had to wait for an I/O operation, CPU would also have to wait until I/O was finished.

CPU requires sophisticated data structures to implement **multiprogramming**. CPU must be able to decide which job is to be executed next. Some jobs will fit into memory at once. Some jobs may have to remain on disk. The operating system must be able to perform some sort of **scheduling** on the jobs in job pool. It determines which jobs will stay on the disk and which ones will be loaded into memory. The operating system must have some form of memory management. **Memory management** is used to keep track of which jobs are stored where and how much space is available.

Sometimes, there may be many jobs that can run at any one time. The interaction between jobs must be controlled. In general, one job should not be able to manipulate the data or program code of another job.

1.3.3 Time Sharing

In **batch system**, the user cannot interact with the job when it is being executed. It means that all possible problems must be anticipated beforehand as the user cannot make corrections during execution. It becomes very difficult when a program has to go through many phases such as compilation, linking etc. It may be difficult to define what to do if a particular phase fails. Another problem is the debugging of a program. All debugging is static. The only information to find out is why the program is giving incorrect output at various stages of execution.

Time sharing was introduced to make computer systems more interactive. CPU is the most important resource that is shared. Each job gets CPU for a small amount of time. When the allotted time period for a job is used, the next job in line is allocated CPU.

The switching between jobs occurs very frequently. It allows the user to interact with the job as it is running. Operating system should enable the user to interact with jobs that are executing. The communication usually occurs via keyboard. The user gets a prompt to enter commands. The user must know the status of the job in order to enter relevant commands. The output of a job is usually presented on a monitor.

Generally, the commands given by the user take very little time to execute. The control returns to command line after finishing a command. It displays a prompt to indicate that the system is ready to execute another command.

1.3.4 Time Sharing Systems

Time sharing system is a multi-user, multi-process and interactive system. It allows multiple users to share computer simultaneously. It is used when several users are linked through communication networks to a single computer. The computer first works on one user's task for a fraction of time. It then goes to the next user's task. It is accomplished through **time slicing**. Each task gets a time slice in **round robin** fashion. The task continues until the time slice ends. The task stops when the given time slice is finished and waits for another time slice.

The computer operates very quickly and can distribute time slices to various tasks. The computer rapidly switches back and forth among different tasks. Time sharing systems use **CPU scheduling** and **multiprogramming** to provide each user with a small portion of time shared computer. The CPU can be allocated to a job only if that job is currently in memory.

A time sharing system is very complex. It is more complex than a multiprogrammed system. It must be able to have several jobs ready in memory simultaneously. It requires some form of memory management and protection. Jobs must be swapped in and out in a reasonable time. It may be achieved by using **virtual memory**. Virtual memory is a technique that allows the execution of a job that may not be completely in the memory.

1.4 Single-Processor Systems

Most systems use a single processor. There are different types of single-processor systems from PDAs to mainframe computers. A single-processor system has one main CPU to execute a general-purpose instruction set including the instructions from user processes. Almost all systems also have other special-purpose processors. They may come in the form of device-specific processors such as disk, keyboard and graphics controllers. They may come in the form of more general-purpose processors in mainframes such as I/O processors to move data quickly among the system components.

The special-purpose processors run a limited instructions set and do not execute user processes. Sometimes they are managed by the operating system. The operating system sends them information about the next task and monitors their status. For example, a disk controller microprocessor receives requests from the main CPU and implements its own disk queue and scheduling algorithm. The main CPU has to perform the disk scheduling. The personal computers have a microprocessor in keyboard to convert keystrokes into codes to send to CPU. The special-purpose processor in some systems are low-level components built into the hardware. The operating system cannot communicate with these processors. They perform their jobs autonomously. The use of special-purpose microprocessors is very common. However, it is still considered a single-processor system not multiprocessor.

1.5 Multiprocessor Systems

Multiprocessing is a type of processing in which two or more processors work together to process more than one program simultaneously. It allows the system to do more work in a shorter period of time. **UNIX** is one of the most widely used multiprocessing systems.

Multiprocessor system is also known as **parallel system or tightly-coupled system**. It means that multiple processors are tied together in some manner. Generally, the processors are in close communication with each other. They share common data structures and a common system clock.

1.5.1 Advantages of Multiprocessor Systems

Some advantages of multiprocessor system are as follows:

1. Reduced Cost	Multiple processors share same resources. Separate power supply or mother board for each chip is not required. This reduces the cost.
2. Increased Reliability	The reliability of system is also increased. The failure of one processor does not affect other processors though it will slow down the machine. Several mechanisms are required to achieve increased reliability. If a processor fails, a job running on that processor also fails. The system must be able to reschedule the failed job or to alert the user that the job was not successfully completed.
3. Increased Throughput	An increase in the number of processes completes the work in less time. It is important to note that doubling the number of processors does not halve the time to complete a job. It is due to the overhead in communication between processors and contention for shared resources etc.

1.5.2 Types of Multiprocessor Systems

There are two main types of parallel processing:

1. Symmetric Multiprocessing

A method of processing in which multiple processors work together on the same task is called **symmetric multiprocessing**. Each processor runs an identical copy of the same operating system. Symmetric multiprocessing treats all processors equally. I/O can be processed on any processor. The processors communicate with each other as needed. It enables many processes to be run at once without degrading performance. Each CPU is separate and it is possible for one CPU to be idle while another CPU is overloaded.

Symmetric multiprocessing is easier to implement in operating systems. It is the method mostly used by the operating systems that support multiple processors. The most common multiprocessor systems today use **SMP architecture**.

2. Asymmetric Multiprocessing

A multiprocessing technique in which each processor is dedicated to a particular task is called **asymmetric multiprocessing**. It has one **master** processor and remainder processors are called **slave**. The master distributes tasks among the slaves. I/O is usually done by the master only. An example is disk management where the responsibility of managing disk activities is not handled by main processor. It is performed by a microprocessor that has its own memory. It only performs disk management tasks such as how to move the head of disk to read data from disk into memory etc.

1.6 Distributed Systems

A system that distributes the computation among several physical processors is called **distributed system**. It is also known as **loosely-coupled** system. It means that the processors do not share memory, data structures or system clock. Each processor has its own local memory. A processor can only communicate with other processors through a communication line usually over a network. The processors in distributed system may vary in size, speed and function. They may include microprocessor, workstations, minicomputers and large general purpose computer system. These processors are given different names such as sites, nodes, computers, machines and hosts etc.

1.6.1 Advantages of Distributed Systems

The main advantages of building distributed systems are as follows:

1. Resource Sharing

Each computer in the distributed system may have specific resources. Other machines may utilize its resources while being part of the distributed system.

2. Computation Speedup

If a computation can be split up into sections, it is possible to give different sections to different machines. Communication is usually fairly expensive since the machines may not be physically close to each other.

3. Reliability

It provides more reliability for different jobs. It is done by providing the replication of resources across the system. The failure of any one computer will not affect the availability of the system's resources. Another sites may be able to continue the processing.

4. Scalability

Scalability is an important advantage of distributed systems. It means that distributed system can grow by adding more computers without affecting the existing applications and users.

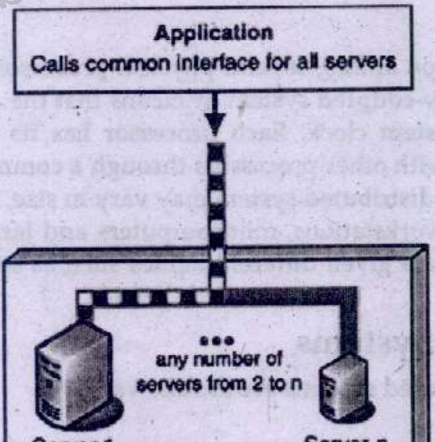
1.7 Clustered Systems

A clustered system uses multiple CPUs to accomplish a task. It is different from parallel system in that clustered system consists of two or more individual systems coupled together. The clustered computers share storage and are closely linked via LAN networking.

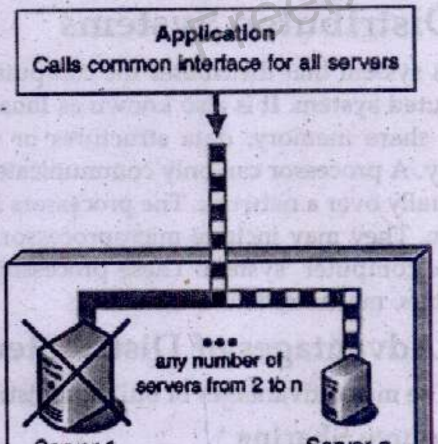
The clustering is usually performed to provide **high availability**. A layer of cluster software runs on cluster nodes. Each node can monitor one or more nodes over the LAN. The monitored machine can fail in some cases. The monitoring machine can take ownership of its storage. The monitoring machine can also restart applications that were running on the failed machine. The failed machine can remain down but the users will see a brief of the service.

The clustered system can be of the following forms:

Asymmetric Clustering In this form, one machine is in hot standby mode and other machine runs the application. The hot standby machine performs nothing. It only monitors the server. It becomes the active server if the server fails.


Application
Calls common interface for all servers
any number of servers from 2 to n
Server 1
Server n
Server Cluster

Normal Condition


Application
Calls common interface for all servers
any number of servers from 2 to n
Server 1
Server n
Server Cluster

Failure Condition

Symmetric Clustering In this form, two or more machines run the applications. They also monitor each other at the same time. This mode is more efficient because it uses all available machines. It can be used only if multiple applications are available to be executed.

Some other forms of clusters are **parallel clusters** and **clustering over WAN**. Parallel clusters allow multiple hosts to access the same data on a shared storage. Most operating systems do not provide the support for simultaneous data access by multiple hosts. That is why parallel clusters are usually accomplished by special versions of software and special releases of applications. An example of such software is **Oracle Parallel Server**. It is a version of Oracle database that is designed to run on parallel clusters. Each machine runs Oracle and a layer of software tracks access to the shared disk. Each machine also has full access to all data in the database.

Most clusters do not support shared access to data on the disk. The distributed file systems must provide access control and locking to the files to prevent conflicting operations. This type of service is commonly known as **distributed lock manager (DLM)**.

1.8 Real-Time Systems

A **real time system** is a system that ensures the exact time requirements for a job. If a real-time system does not complete a task in a certain time, it may cause a breakdown of the entire system it is running. Some applications require to be serviced within a time period:

- Industrial control systems
- Automobiles (brakes)
- Airplane guidance
- Medical devices

1.8.1 Types of Real-Time Systems

There are two types of real time systems:

1. Hard Real Time Systems

A **hard real time system** guarantees that a job will complete within a specified time period. This system must ensure that all delays on processing, input and output are bounded. The system cannot wait indefinitely so the hard time systems are usually very limited. There is generally no secondary storage such as disk drives as a disk request can take a variable time to process.

Some examples of a hard real time system are the software that run the autopilot in a jumbo jet or the imaging software on a missile.

2. Soft Real Time Systems

A **soft real time system** is a much less restrictive version of a hard real time system. A soft real time system does not guarantee that a job will complete within a specified time period. However, it tries its best to finish the job as soon as possible. If a critical real time job enters the system, the operating system may assign the highest priority to that task and execute it continuously until it completes.

The lack of a guarantee makes a soft real time system more limited in its application for industrial activities. Soft real time systems are used in multimedia, virtual reality etc. The operating systems that provide soft realtime support are Windows NT/2000, Linux, Solaris.

1.9 Hand-Held Systems

Hand-held systems include a **personal digital assistant** or **PDA**. It is a remarkable, tiny, fully functional computer that can be held in one hand. PDA can hold downloaded email and play music.

The small computer organizers were available in 1980s. The idea of making hand-held computer for storing addresses and phone numbers, taking notes and keeping track of daily appointments originated in 1990s. One of the first PDAs that was commercially available was **Apple Computer's Newton Message Pad**. It was too big, expensive and complicated. Its handwriting recognition program was poor.

In 1996, the original **Palm Pilot** was introduced and was a hit with consumers. It was small and light enough to fit in a shirt pocket. PDA was originally intended to be simple digital calendars. It has evolved into machines for playing games or music and downloading information from the Internet. It complements a desktop or laptop computer.

1.10 Multimedia Systems

Most operating systems are designed to handle conventional data such as text files, programs, word processing documents and spreadsheets. However, the multimedia data is also being used in computer systems nowadays. Multimedia data consists of audio, video and the conventional files. This data is different from conventional data because it must be delivered according to certain time restrictions in some cases. For example, a video may have the restriction to be delivered 30 frames per second etc.

Multimedia applications include MP3 DVD movies, video conferencing and video clips of movie or news stories downloaded over the Internet. They also include broadcasting of speeches or sport events over the Internet. A typical multimedia application is a combination of audio and video. For example, a movie consists of separate audio and video tracks. These applications are not only available for desktop computers but also for smaller devices such as PDA and cellular telephones etc.

1.11 Interrupts

An **interrupt** is a hardware-generated **change-of-flow** within the system. An **interrupt handler** deals with the cause of the interrupt. The control is then returned to the interrupted context and instruction.

1.11.1 Common Functions of Interrupts

Each computer architecture has its own interrupt mechanism but they all have several functions in common. Different functions of interrupts are as follows:

- When an interrupt occurs, the control is transferred to the interrupt service routine. The interrupt service routine is generally accessed through an interrupt vector. Interrupt vector knows where to find the appropriate interrupt service routine for the current interrupt.
- The interrupt architecture must save the address of the instruction that has been interrupted (the *program counter*).
- Incoming interrupts must be disabled if there is an interrupt currently being processed. This is to prevent interrupts from being lost or overwritten by newly arriving interrupts.
- An interrupt is generally caused by a hardware device such as a device driver allowing the CPU to know that some data has been read from the device and is waiting in the local device buffer
- A trap is a software-generated interrupt that is either caused by an error or by a user request for an I/O operation.
- An operating system is interrupt driven. This means that if there are no interrupts, then the system will be idle.
- The operating system must preserve the state of the CPU by storing the contents of the registers and the program counter.
- The operating system must determine which type of interrupt has occurred. This can be done either by polling or by using a vectored interrupt system. Polling is the

systematic checking of each device to see if it was the device responsible for generating the interrupt. If the operating system has a vectored interrupt system, then the identity of the device and the type of interrupt will be easily identifiable without checking each device.

- The operating system must provide a segment of code that specifies what action is to be taken in the event of an interrupt. There must be a code segment that is specific to each type of interrupt.

1.11.2 Classes of Interrupts

Different classes of interrupts are as follows:

1. Supervisor Call Interrupts (SVC)

These interrupts are initiated by a running process that executes SVC instruction. It is a user request for a particular system service such as performing I/O or obtaining more storage etc.

2. I/O Interrupts

These are initiated by I/O hardware. These interrupts send signal to CPU to indicate that the status of I/O device has changed. I/O interrupts occur when I/O operation completes, I/O error occurs or a device gets ready etc.

3. External Interrupts

These are caused by different events. These events can the expiration of a quantum on interrupting clock, pressing of interrupt key or receipt of a signal from another processor in a multiprocessor system etc.

4. Restart Interrupts

These interrupts occur when the restart button of console is pressed or restart signal is received from another processor in multiprocessor system.

5. Program Check Interrupt

These are caused by different problems during the execution of machine language instructions. Some problems include division by zero, arithmetic overflow or underflow, wrong format of data etc.

6. Machine Check Interrupts

These are caused by the hardware that does not function properly.

1.12 Operating System Modes

The operating system code and user-defined code must be distinguished to ensure the proper execution of the operating system. Most of the computer systems provide hardware support to differentiate among various modes of execution.

The two separate modes of operation include **user mode** and **kernel mode**. The kernel mode is also known as **supervisor mode, system mode or privileged mode.**

A system enters kernel mode in the following three ways:

- A special instruction called **supervisor call** or **system call** can be used to set the system in kernel mode.
- A **trap** can be used to set the system in kernel mode.
- An **interrupt** can be used to set the system in kernel mode.

A **mode bit** is added to the hardware of the computer to indicate the current mode. A value of 0 indicates kernel mode and a value of 1 indicates the user mode. The mode bit is used to distinguish between the tasks executed by the operating system and the user. The system is in user mode when the computer executes on behalf of a user application. It changes to kernel mode if the user application requests a service from the operating system via system call.

The hardware starts in kernel mode at the time of system boot. The operating system is loaded and it starts the user applications in user mode. The hardware switches from user mode to kernel mode when a trap or interrupt occurs. Thus, the system is in kernel mode when the operating system controls the computer. The system always switches to the user mode before passing the control the user program.

The dual mode of operation ensures the protection of the operating system from users and one user from other users. This protection is accomplished with designating some possibly harmful instructions as **privileged instruction**. The hardware allows the execution of privileged instructions only in kernel mode. If an attempt is made to execute a privileged instruction in user mode, the hardware does not execute it and takes it as illegal. System calls are used by user program to request the operating system to perform some tasks on the behalf of user program. A system call can be used in different ways.

1.13 I/O Structure

A general-purpose computer consists of central processing unit and different device controllers connected through a common bus. Each device controller manages a specific type of device. A device controller maintains a local buffer storage and a set of special-purpose registers. The device controller is responsible to move the data between peripheral devices and its local buffer storage.

The main types of responses to I/O interrupts are as follows:

1. Synchronous I/O

The I/O operation is started and the control is returned to the user process when it is completed. It can be accomplished by making the CPU idle using special wait instruction or using wait loop until the next interrupt.

2. Asynchronous I/O

The control returns to the user process without waiting for I/O operation to complete. In this case, a specific system call is required to allow the user program to wait for I/O completion. It also requires a device-status table containing an entry for each I/O device with information about its current status. A wait queue is required as other processes may also issue requests to the same device being used.

Direct memory access (DMA) is used for high-speed I/O devices. The device controller transfers an entire block of data directly to or from its own buffer storage to memory without any intervention by CPU.

1.14 Operating System Components

An operating system has many components to manage all resources of computer system while ensuring proper execution of programs.

1.14.1 Process Management

A process is a program in execution. A process needs certain resources, including CPU time, memory, files and I/O devices in order to accomplish its task. The operating system is responsible for the following activities related to process management:

- Creation and deletion of user and system processes.
- Suspension and resumption of processes.
- Mechanisms for process synchronization.
- Mechanisms for process communication.
- Mechanisms for handling of deadlock situations

1.14.2 Main Memory Management

Memory management is a process of optimizing the use of main memory. RAM is used to store data and instructions temporarily during execution. Operating system allocates memory area to different programs. The allocated memory area is deallocated when the program finishes. Some operating systems allocate a part of storage medium like hard disk as additional RAM. A part of a running program can be in RAM and the remaining part may be on hard disk. This is known as **virtual memory**.

The area of hard disk used for virtual memory is called **swap file**. It is used to exchange data, information, and instructions between memory and hard disk. The amount of data and program instructions that can swap at a given time is called **page**. The process of swapping items between memory and hard disk is called **paging**.

A situation in which most of the time of operating system is wasted in paging instead of executing the program is called **thrashing**.

The operating system is responsible for the following activities in connection with memory management:

- Keeping track of which parts of memory are currently being used and by whom.
- Deciding which processes to load when memory space becomes available.
- Deciding how much memory is to be allocated to a process.
- Allocating memory space as needed.
- De-allocating memory space as needed.

1.14.3 Secondary Storage Management

Main memory or primary storage is volatile and too small to accommodate all data and programs permanently. The computer system must provide secondary storage to back up main memory. Most modern computer systems use disks as the principle storage medium for both programs and data. The operating system is responsible for the following activities in connection with disk management:

- Free space management
- Storage allocation and de-allocation
- Disk Scheduling

1.14.4 I/O System Management

The role of operating system in I/O system is to manage and control I/O operations and I/O devices. One goal of I/O is to provide the simplest interface and to optimize I/O for maximum concurrency. The I/O system manages different I/O devices such as mouse, keyboard, monitor and printer etc.

The I/O system consists of the following:

- A buffer caching system
- A general device driver interface
- Drivers for specific hardware devices

1.14.5 File Management

A file is a collection of related information defined by its creator. A file can be a program like word processing program. It can be a data file like images, songs etc. A file management system is a system software that provides services to the user and applications related to the use of file. The operating system is responsible for the following activities in connection with file management:

- File creation
- File deletion
- Directory creation
- Directory deletion
- Support of primitives for manipulating files and directories
- Mapping files onto secondary storage
- File backup on stable (non-volatile) storage media

1.14.6 Protection System

If a computer system has multiple users and allows concurrent execution of multiple processes, the processes must be protected from each other's activities. The **protection** refers to a mechanism to control access by programs, process or users to the system and resources.

The protection mechanism must:

- Distinguish between authorized and unauthorized usage
- Specify the controls that are to be imposed
- Provide a means of enforcement

1.14.7 Networking (Distributed Systems)

A **distributed system** is a collection of processors that do not share memory or a clock. Each processor has its own memory. The processors in the system are connected through a communication network. A distributed system provides access to various system resources.

The access to a shared resource allows:

- Computation speed-up
- Increased data availability
- Enhanced reliability

1.14.8 Command Interpreter System

Command interpreter is an interface between the user and operating system. It is one of the most important system programs for operating system. It is used to read and execute user commands. Some operating systems provide command interpreter in the kernel. The command interpreter system is also known as **command-line interpreter** or **shell**. Many commands are given to operating system by control statements which deal with following:

- Process creation and management
- I/O handling
- Secondary storage management

- Main memory management
- File-system access
- Protection
- Networking

The program that reads and interprets control statements has many different names. Some of these are:

- Control card interpreter
- Command line interpreter
- Shell (in Unix)

The function of this program is to get and execute the next command statement.

1.15 Kernel

The **kernel** provides the most basic interface between the computer itself and the rest of the operating system. The kernel is responsible for the management of the central processor. The kernel includes the **dispatcher** to allocate the central processor, to determine the cause of an interrupt and initiate its processing, and some provision for communication among the various system and user tasks currently active in the system.

Kernel is the core of an operating system. The main functions of kernel are as follows:

- It provides a mechanism for creation and deletion of process.
- It provides CPU scheduling, memory management and device management for these processes.
- It provides synchronization tools so that the processes can synchronize their actions.
- It provides communication tools so that processes can communicate with each other.

Scheduling is a fundamental operating system function. Almost all computer resources are scheduled before use. CPU is one of the primary computer resources. Its scheduling is often performed in the operating system.

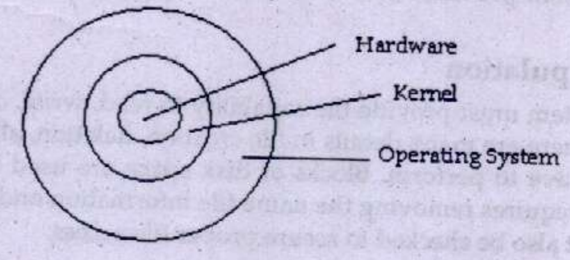


Figure: Structure of kernel-based operating system

The kernel-based design is often used to design operating system. The kernel (also called **nucleus**) is a set of primitive facilities over which the rest of operating system is built.

A kernel provides an environment to build an operating system in which the designer has considerable flexibility. The policy and optimization decisions are not made at kernel level. An operating system is an orderly growth of software over kernel where all decisions regarding process scheduling, resource allocation, execution environment, file system and resource protection etc. are made.

The kernel only supports the notion of processes. It does not include the concept of a resource. Operating systems have matured in functionality. and complexity and more functionality has been relegated to the kernel. A kernel should contain a minimal set of functionality that is adequate to build an operating system with a given set of objectives.

1.16 Operating System Services

An operating system provides an environment for the execution of programs. It provides certain services to programs and to the users of these programs. These services are provided for the convenience of the programmer to make the programming task easier.

Different services provided by the operating system are as follows:

1. User Interface

All operating systems provide a user interface. The user interface can be of different types such as command-line interface, batch interface and graphical user interface. The **command-line interface** uses text commands and a method for entering them. The **batch interface** allows the user to enter commands into files and then execute these files. The **graphical user interface (GUI)** is a window system with pointing device to give instructions, select menus and make selections etc. It is the most popular type of user interface.

2. Program Development

Operating system provides different services for the programmers to create programs such as editors and debuggers. These services are typically available as utility programs. The services are known as program development tools.

3. Program Execution

Operating system provides the services to execute programs. It loads the instructions and data into memory, initializes the files and I/O devices and prepares other resources.

4. Access to I/O Devices

The operating system provides an interface to the programmer to access I/O devices easily.

5. File System Manipulation

The operating system must provide the capability to read, write, create and delete files from a user program. There are many details in file creation, deletion, allocation, and naming that users should not have to perform. Blocks of disk space are used by files and must be tracked. Deleting a file requires removing the name file information and freeing the allocated blocks. Protections must also be checked to assure proper file access.

6. Error Detection

Operating system provides the facility of error detection and response. It is the ability to detect errors in CPU, memory hardware, I/O devices or in user programs. Error detection occurs at both hardware and software levels. At hardware level, all data transfers must be checked to ensure that data have not been corrupted or changed. At software level, media must be checked for data consistency. For example, the number of allocated and unallocated blocks of storage should match the total number of blocks on the device.

Some additional operating system services that do not help the user but ensure that the systems run efficiently. These services are as follows:

Communications	It is the exchange of information between processes that may be executing on the same computer or on completely different computers. It is usually implemented via shared memory or message passing.
Resource Allocation	It is the ability to allocate resources to multiple users or to multiple jobs running at the same time.
Accounting	It is the ability to keep track of and to record which users use how much and what kinds of computer resources.
Protection	It is an ability to ensure all access to system resources is controlled.

1.17 System Calls

System calls provide an interface between a user program and operating system. The system calls exposes the services offered by the operating system to user programs. These are set of functions/methods. Most system calls are written in assembly language and are machine dependent.. Several higher level languages such as C also allow to make system calls directly.

The system call interface layer contains entry point in kernel code. All system resources are managed by the kernel. Any request from user or application that involves access to any system resource must be handled by kernel code. The user process must not be given open access to kernel code for security reason. Many opening into kernel code called system calls are provided to user so that the user processes can invoke the execution of kernel code. System calls allow processes and users to manipulate system resources.

There are three general methods that are used to pass information (parameters) between a running program and the operating system.

- One method is to store parameters in registers.
- Another is to store parameters in a table in memory and pass the address of table.
- The third method is to push parameters on stack and allow operating system to pop the parameters off the stack.

1.17.1 Main Types of System Calls

The main types of system calls are as follows:

Process Control	These types of system calls are used to control the processes. Some examples are end, abort, load, execute, create process, terminate process etc.
File Management	These types of system calls are used to manage files. Some examples are Create file, delete file, open, close, read, write etc.
Device Management	These types of system calls are used to manage devices. Some examples are Request device, release device, read, write, get device attributes etc.
Information Maintenance	These types of system calls are used to maintain information. Some examples are Get time or date, set time or date, get system data, set system data etc.
Communications	These types of system calls are used to for communication. Some examples are Create, delete communication connection, send, receive messages etc.

Review Questions

1. What are the two main purposes of an operating system?

The main purpose of an operating system is to provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner. Another purpose is to allocate the separate resources of the computer as needed to solve problems. The allocation process should be as fair and efficient as possible.

2. What are the basic functions of operating system?

Operating system controls and coordinates the use of hardware among the various application programs for various uses. Operating system acts as resource allocator and manager. There may be many conflicting requests for resources. The operating system must decide which requests are allocated resources to operate the computer efficiently and fairly. Operating system is also controls the user programs to prevent errors and improper use of computer. It is especially concerned with the operation and control of I/O devices.

3. What is the fundamental goal of computer systems?

The fundamental goal of computer systems is to execute user programs and solve user problems.

4. What is a simple batch system?

In simple batch system, a professional computer operator groups jobs by characteristics and runs groups of similar jobs together efficiently.

5. Write the essential properties of batch operating systems.

Jobs with similar needs are batched together and run through the computer as a group by an operator or automatic job sequencer. Performance is increased by attempting to keep CPU and I/O devices busy at all times through buffering, off-line operation, spooling, and multiprogramming. Batch is good for executing large jobs that need little interaction.

6. In what ways are batch systems inconvenient for users?

The users cannott interact with their jobs to fix problems. They must anticipate problems or else debugging could be a mess with machine-language dumps. There may also be long turn around times.

7. What is "automatic job sequencing"?

System proceeds from one job to the next without human intervention.

8. What were control cards used for?

They were used by monitors to know the required resources for current job. They are used to know when to use these resources and with which file. They tell monitor when it reaches end of job.

9. Describe one kind of off-line operations.

An example is cards copied to magnetic tape. It was then mounted on computer. The output of computer was dumped to magnetic tape that which was mounted for output to a printer.

10. What were the advantages of off-line operations?

The main computer no longer constrained by speed of card reader. The application programs used logical I/O devices instead of physical I/O devices. Programs did not have to be rewritten when new I/O devices replaced old ones.

11. What is multiprogramming?

Multiprogramming is the rapid switching of the CPU between multiple processes in memory. It is commonly used to keep the CPU busy while one or more processes are doing I/O.

12. What is the main advantage of multiprogramming?

Multiprogramming makes efficient use of CPU by overlapping the demands for CPU and its I/O devices from various users. It increases CPU utilization.

13. What is the difference between multiprogramming and multitasking?

Multitasking is a logical extension of multiprogramming. The CPU executes multiple jobs by switching among them. The switches occur so frequently that the users can interact with each program with each program it is running.

14. What is the difference between multiprogramming and multiprocessing?

A multiprogramming system runs more than one program simultaneously on one processor. The system attempts to keep several programs resident in main memory and switches the processor rapidly between them.Multiprogramming was developed to improve processor and I/O resource utilization.A multiprocessor is a computer system with more than one processor.Multiprocessing was developed in an effort to increase processing speeds by allowing truly parallel computation.

15. What is the difference between a multiprocessor and a multiprocessing system?

Multiprocessor refers to a system with two or more processors or CPUs. Multiprocessing refers to a system that can process one or more tasks at a time.

16. Give an example when it is preferred to use a multiprocessing system.

Multiprocessing is required when the user want to run two programs at the same time. For example, a report can be printed while listening to the music etc.

17. What are the advantages and disadvantages of multiprocessor systems?

Multiprocessors can save money by not duplicating power supplies, housings, and peripherals. They can execute programs more quickly and can have increased reliability. They are also more complex in both hardware and software than uniprocessor systems.

18. What is the difference between symmetric and asymmetric multiprocessing?

Symmetric multiprocessing treats all processors equally and I/O can be processed on any CPU. Asymmetric multiprocessing has one master CPU and the remainder CPUs are slaves. The master distributes tasks among the slaves and I/O is usually done by the master only.

19. What are the main differences between the operating systems for mainframe computers and personal computers?

The resources in operating system for mainframe computer system are utilized effectively. But level of user interaction with the computer system is low. The operating systems for personal computers maximize user convenience and responsiveness instead of maximizing CPU and peripheral utilization.

20. What do you mean by online?

It simply means that a computer is turned on and directly connected to the network. For example, when on millions of users are online because their PCs are connected to the Internet.

21. Write the important features of time-sharing operating systems.

Time sharing was introduced to make computer systems more interactive. CPU is the most important resource that is shared. Each job gets CPU for a small amount of time. When the allotted time period for a job is used, the next job in line is allocated the CPU.

22. How is time-sharing usually implemented?

Each user is given a brief time-slice for her job in round-robin fashion. The job continues until the time-slice ends. Then her job stops until it is her turn again.

23. When should a user use time-sharing system than personal computer or single-user?

Time-sharing is useful when there are few other users or the task is large and the hardware is fast. The full power of the system can be brought to bear on the user's problem. The problem can be solved faster than on a personal computer. Another case occurs when lots of other users need resources at the same time. A personal computer is best when the job is small enough to be executed reasonably on it and when performance is sufficient to execute the program to the user's satisfaction.

24. The computer in a time-sharing system is shared among many users. How does each user feel that it is dedicated to him?

Interactive I/O typically may take a long time to complete. Input may be bounded by the user's typing speed. CPU does not sit idle when interactive input takes place. The operating system rapidly switches CPU to the programs of some other users. The system switches rapidly from one user to the next. Each user feels that the entire computer system is dedicated to him.

25. Define the term interactive computing.

Interactive computing involves a user being present when jobs or processes are run and carrying on a dialog with the machine. Interactive computing ordinarily involves a series of short user requests that require prompt service.

26. What is the important aspect of a real-time system?

A real time operating system has well defined fixed time constraints. The process must be done within the defined constraints or the system will fail. An example is the operating system for a flight control computer or an advanced jet airplane.

27. What is the difference between hard real-time and soft real-time system?

A hard real time system guarantees that a job will complete within a specified time period. This system must ensure that all delays on processing, input and output are bounded. A soft real time system is a much less restrictive version of a hard real time system. It does not guarantee that a job will complete within a specified time period. However, it tries its best to finish the job as soon as possible.

28. What are main differences between batch multiprogramming and time-sharing system?

The main objective of batch multiprogramming system is maximizing processor use. The main objective of time-sharing system is to minimize the response time. Time sharing systems support interactive computing by responding to commands entered at the terminal. Batch multiprogramming does not support interactive computing.

29. What is the difference between clustered system and multiprocessor system?

The clustered system consists of two or more individual systems that are coupled together and share the storage space. They are also closely attached through a network. The multiprocessor systems can be a single physical entity comprising of multiple CPUs. A clustered system is less tightly coupled than a multiprocessor system.

30. What are the features of operating systems for embedded environment?

Embedded systems have very specific tasks. The embedded operating system runs commonly on primitive, lacking advanced features hardware. Thus operating systems in this situation also provide limited features. They have little or no user interface and designed especially to monitor and manage hardware devices.

31. Differentiate between a loosely coupled system and a tightly coupled system?

Multiprocessors are typically tightly coupled but distributed workstations on a network are not. In a tightly-coupled system, the delay experienced when a message is sent from one computer to another is short. The data rate is high. The number of bits per second that can be transferred is large. In a loosely coupled system, there is longer delay when message is transferred from computer to another. The data rate is also low. Two CPU chips on the same circuit board and connected by wires are tightly coupled. Two computers connected by modems over the telephone system or LAN are loosely coupled.

32. Differentiate between a trap and an interrupt?

An interrupt is a hardware-generated change-of-flow within the system. An interrupt handler deals with the cause of interrupt. The control then returns to the interrupted context and instruction. A trap is a software-generated interrupt. An interrupt can be used to signal the completion of an I/O. A trap can be used to call operating system routines or to catch arithmetic errors.

33. How is an interrupt executed?

The I/O driver sends a signal through a special interrupt line to the CPU when it has finished with an I/O request.

34. Give three examples of interrupt.

Three examples of interrupt include I/O interrupt, program check interrupt and machine check interrupt.

35. What is ISR?

Interrupt Service Routine or Interrupt Handler is a program. It is part of operating system. It determines the nature of the interrupt and performs whatever actions are needed.

36. What is an interrupt vector?

A list giving the starting addresses of each interrupt service routine.

37. What is a software interrupt? How does it work?

Software may trigger an interrupt by executing a special operation called a system call. A system call is treated by the hardware as software interrupt. Control passes through the interrupt vector to a service routine in the operating system and the mode bit is set to monitor mode. The system call service routine is part of operating system.

38. Why is it considered that modern operating systems are interrupt driven?

An operating system will do nothing if there are no processes to execute, no I/O devices to service and no users to respond. It will wait for something to happen. Events are almost always signaled by the occurrence of an interrupt or a trap.

39. What is kernel?

The kernel is a portion of the operating system that includes the most heavily used portions of software. Generally, the kernel is maintained permanently in main memory. The kernel runs in a privileged mode and responds to calls from processes and interrupts from devices.

40. What is difference between user mode and kernel mode?

These are the execution modes in an operating system. The user applications run in user mode and core operating system components run in kernel mode. Kernal mode provides access to any resource regarding the hardware. The user mode can only access the user resources.

41. What is the purpose of command interpreter? Why is it usually separate from kernel?

Command interpreter is used to read commands from the user or file of commands. It executes them usually by turning them into one or more system calls. It is usually not part of the kernel since the command interpreter is subject to changes.

42. What is the purpose of system calls?

System calls allow user-level processes to request services of the operating system.

2

PROCESS MANAGEMENT

Chapter Overview

- 2.1 Process
    - 2.1.1 Process Attributes
- 2.2 Process States
- 2.3 State Transition in Process
- 2.4 Process Control Block
- 2.5 Process Scheduling
    - 2.5.1 Scheduling Queues
    - 2.5.2 Scheduler
    - 2.5.3 Context Switch
- 2.6 Operations on Processes
    - 2.6.1 Process Creation
    - 2.6.2 Process Termination
- 2.7 Inter-Process Communication (IPC)
    - 2.7.1 Independent Processes
    - 2.7.2 Cooperating Processes
    - 2.7.3 Shared Memory Model
    - 2.7.4 Message Passing Model
        - 2.7.4.1 Naming
        - 2.7.4.2 Synchronization
        - 2.7.4.3 Buffering

Review Questions

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

2.1 Process

A process is a program in execution. It is the basic unit of resource allocation in an operating system. A process has a starting and ending point. It must execute and progress sequentially. A process is controlled and scheduled by operating system. An operating system executes different programs depending on the system.

A program is a passive entity. It specifies the logic of data manipulation and I/O actions. A program does not perform the actions itself. A process is an active entity that uses system resources. It performs the actions specified in the program.

2.1.1 Process Attributes

Imp.

A process has the following attributes:

Program counter	It is used to keep track of the address of the next instruction to be executed
Stack	A stack is used to store temporary data for the process
Data section	A data section is used that may contain global variables
Heap	A process may include a heap. It is the memory that is dynamically allocated during process execution.

2.2 Process States

Process state describes the nature of current activity in a process. The state of a process changes as it executes. A process may be in one of the following states:

New	The process is being created and is not yet in memory
Running	The process has an instruction that is being executed
Waiting (Blocked)	The process is waiting for some event to occur
Ready	The process is waiting to be executed
Terminated execution	The process has finished execution and is no longer in memory

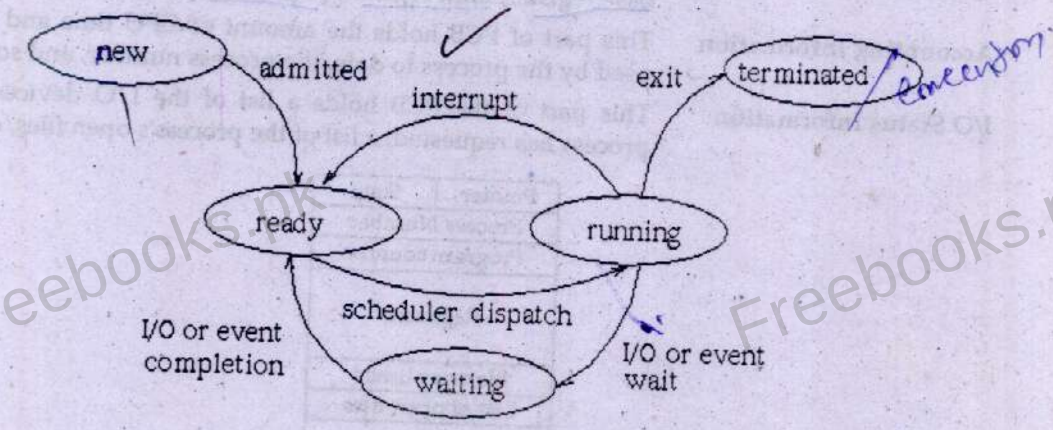


Figure: Five State Process Model

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

2.3 State Transition in Process

Different state transitions in a process can be as follows:

New → Ready	The process is admitted to ready queue and can be considered by CPU scheduler.
Ready → Running	CPU scheduler chooses that process to execute next according to some scheduling algorithm.
Running → Ready	Process has used its current time slice.
Running → Blocked	Process is waiting for some event to occur such as an I/O operation to complete.
Blocked → Ready	The event has occurred for which the process was waiting.

2.4 Process Control Block

Process control block is a data structure. It contains all information for a process to be used for controlling its execution. It also contains the information related to the resource allocated to the process. Every process has its own PCB. The process control block is very important. It should be stored in protected area where the user cannot access it.

The information that must be saved usually consists of the following:

Process State	It indicates the information about the process. The state can be ready, waiting, etc.
Program Counter	It points to the next instruction to be executed when the process gains the CPU.
CPU Register	The contents of CPU register for the process. All the temporary data that will be destroyed by next process that gets the CPU after an interrupt occurs.
CPU Scheduling	It holds information needed for CPU scheduling, such as process priority and any scheduling-queue pointers
Memory Management	It is the information required by the memory manager such as base register, limit register, page table for the process etc.
Accounting Information	This part of PCB holds the amount of CPU time and real time used by the process to date, the process number, and so forth.
I/O Status Information	This part of the PCB holds a list of the I/O devices that the process has requested, a list of the process's open files, etc.

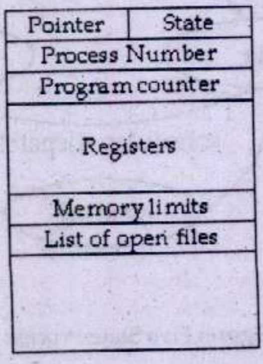


Figure: Process Control Block

2.5 Process Scheduling

Process scheduling is a technique that is used when there are limited resources and many processes are competing for them. Multiprogramming tries to ensure that there is some process running at all times. This is done to utilize the CPU as much as possible. In timesharing system, the CPU switches so frequently between jobs that the user does not feel that the machine is being shared by many processes or even many users.

If the system has more than one processor, then it is possible to execute more than one process at the same time. In a single processor system, only one process can be executed at any given time. If there are more processes than processors, then the operating system must schedule the processes. It means that some processes will be executed and others will have to wait. There are many strategies for deciding which process should be assigned the CPU.

2.5.1 Scheduling Queues

The most common types of queues and their purpose are as follows:

Job Queue	Each entering process goes into job queue. The processes in job queue reside on secondary storage and await the allocation of main memory.
Ready Queue	The set of all processes that are in main memory and are waiting for CPU time are kept in ready queue.
Waiting Queues	The set of processes waiting for allocation of certain I/O devices, are kept in waiting queue.

Process Migration between Various Queues

During its lifetime, a process begins in the job queue. It moves between the ready, waiting and possibly job queues and eventually finishes the task. This process of moving between various queues is known as **process migration**.

18

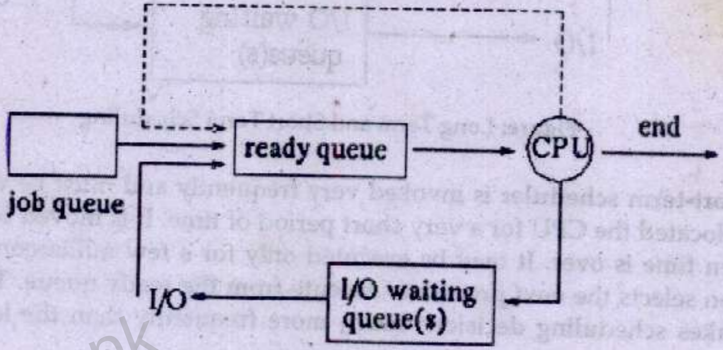


Figure: Process migration between various queues

2.5.2 Scheduler

A process is placed in different scheduling queues throughout its lifetime. The operating system must select the processes from these queues in some way for scheduling purposes. The part of the operating system related to this decision is called the **scheduler.** The algorithm used by scheduler is called the **scheduling algorithm.**

The operating system must perform three different kinds of scheduling:

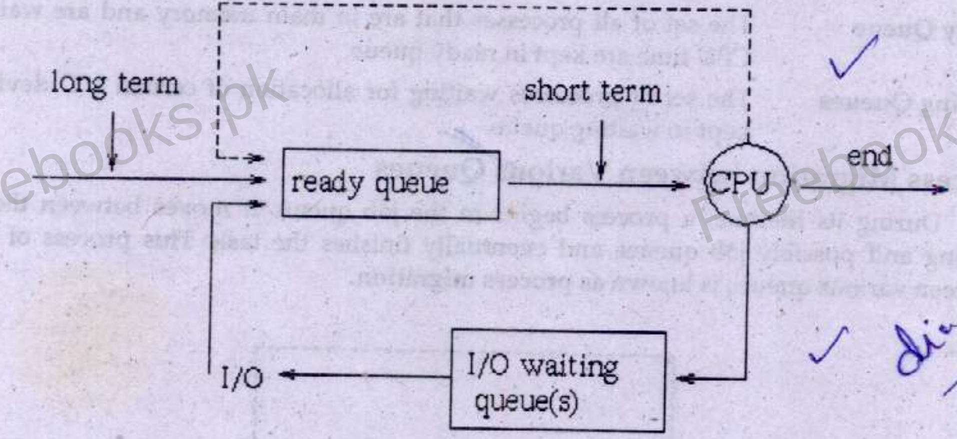
1. Long Term Scheduler

Long term scheduler or **job scheduler** is used to select a job or user program from job queue to the ready queue in main memory. The job becomes a process after it has been added to the ready queue in memory. The long term scheduler controls the number of processes in memory also known as the **degree of multiprogramming**. The system may slow down if too many processes are loaded into memory. However, too few processes in the memory means that the system is not being utilized efficiently.

The average rate of process creation must be equal to the average departure rate of processes leaving the system if the degree of multiprogramming is stable. The long term scheduler may be invoked only when a process leaves the system. It then decides the next process to be executed.

2. Short Term Scheduler

The **short term scheduler** is also known as **CPU scheduler.** It selects a process from ready queue and allocates CPU to that process for execution.



diagram

Figure: Long Term and Short Term Scheduling

The **short-term scheduler** is invoked very frequently and must be very fast. A process is typically allocated the CPU for a very short period of time. It is moved back to ready queue after the given time is over. It may be executed only for a few milliseconds. The short-term scheduler then selects the next process to execute from the ready queue. Thus the short-term scheduler makes scheduling decisions much more frequently than the long-term and mid-term schedulers.

3. Medium Term Scheduler

The **medium-term scheduler** removes processes temporarily from main memory and places them on secondary memory such as a disk drive. It moves it back to main memory when required. It is commonly known as **swapping out** and **swapping in**. The medium-term scheduler may decide to swap out a process which has not been active for some time or it has a low priority. It can also swap out a process that is taking up a large amount of memory to free up main memory for other processes. It swaps in the process later when more memory is available or when the process is unblocked and is not waiting for a resource. It improves the performance of the system by reducing the degree of multiprogramming.

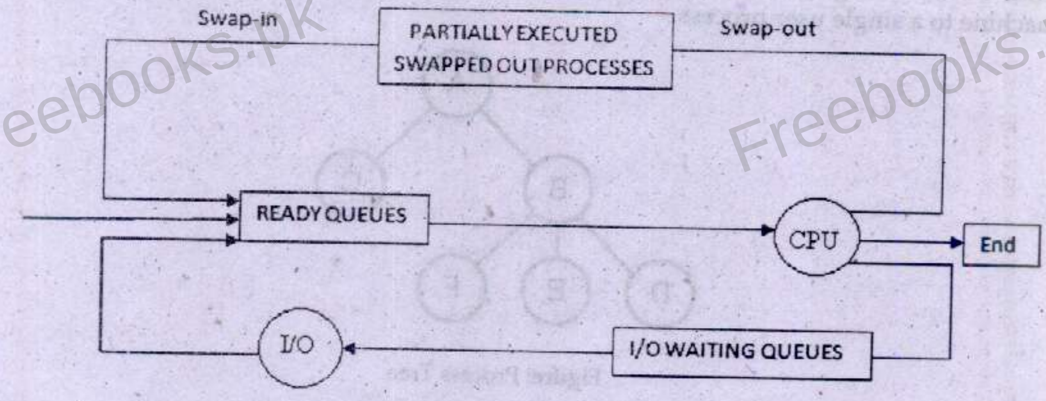


Figure: Addition of Medium-term scheduling

2.5.3 Context Switch

CPU switches from one process to another frequently in multiprogramming systems. The interrupt also cause the CPU to stop the execution of current process and move to another process or kernel routine. In this situation, the system must save the current context or state of the process being executed. This information is used when the same process is executed again later. The information includes the values of CPU registers, process state and memory management information. It is stored in the process control block. This process of switching from one process to another and saving the state of current process is known as **context switch**. The kernel saves the context or state of the old process in its PCB and loads the saved context of the new process. It is important for proper execution of processes. The new process may use the data from the old process if context switch does not occur.

The CPU does not perform any useful work whenever a context switch is performed. The time required by a context switch depend on the hardware support for the switching such as multiple sets of registers etc.

2.6 Operations on Processes

The operating system is responsible for providing a mechanism to create and destroy processes dynamically.

2.6.1 Process Creation

Parent process creates children processes. The children processes can also create other processes forming a tree of processes.

A process may create zero or more new child processes. Process creation may be limited by the amount of resources that a parent process has. If a parent process creates a child process, the child process may create child processes of its own. It results in a tree of processes.

When a child process is created, it may be able to obtain resources directly from the parent. It may only be able to obtain a subset of the parent's resources or it must acquire its own resources. It is usually a good idea to limit the resources of child process as a single user process could create hundreds of child processes. Each might be able to acquire resources of their own, leaving no resources for other processes and effectively giving control of the machine to a single user process.

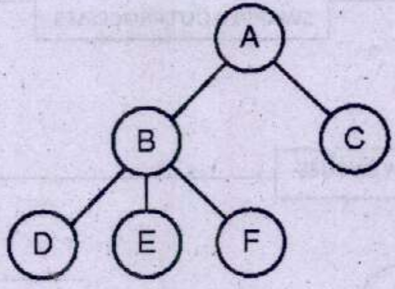


Figure: Process Tree

When a process creates a new process, the parent may continue to execute concurrently with its children or it must wait until some or all its children have terminated execution.

In terms of address space, the child may be an exact duplicate of the parent. This would allow the child and parent to communicate very easily as they share the same address space. It should be possible to load a new program into the address space of the child. It is also possible to create an entirely new process and place a new program into it. This makes it more difficult for the child and parent to communicate since they have nothing in common.

2.6.2 Process Termination

A process normally terminates when it has finished executing its last statement. It typically tells the operating system by issuing some form of the exit() system call. At this point, the process may output some data to its parent if it is a child process. Eventually the process has all its resources removed from it and is no longer part of the job pool.

A child process may be terminated by its parent using the **abort ()** system call. This may occur because the child has asked for too many resources, the child is no longer wanted or the parent is terminating. The operating system may or may not allow a child process to continue executing after its parent terminates. In some cases, the child process should not continue. In other cases, a child process may continue after the parent has finished.

2.7 Inter-Process Communication (IPC)

Interprocess communication (IPC) is a capability supported by operating system that allows one process to communicate with another process. The processes can be running on the same computer or on different computers connected through a network. IPC enables one application to control another application, and for several applications to share the same data without interfering with one another. IPC is required in all multiprogramming systems, but it is not generally supported by single-process operating systems such as DOS. The OS/2 and MS-Windows support an IPC mechanism called **Dynamic Data Exchange**.

Inter-process communication is a mechanism that is used by the processes to exchange data and information. Many processes may be executing at the same time in the operating system. These processes can be divided into two types known as **independent processes** and **cooperating processes**.

end

2.7.1 Independent Processes

A process that cannot affect or be affected by any other process executing in the system is called **independent process**. An independent process does not share data with any other process.

2.7.2 Cooperating Processes

A process that can affect or be affected by any other process executing in the system is called **cooperating process**. A cooperating process shares data with other processes.

Different advantages of process cooperation are as follows: /// End This Chapter

1. Information Sharing

Process cooperation is necessary because many users may need to access the same information such as a shared file etc. An environment must be provided where such information can be accessed by multiple users concurrently.

2. Computation Speedup

One task can be divided into multiple tasks to execute it faster. All divided subtasks can be executed concurrently to speedup the computation. In this case, these subtasks may required to cooperate with each other as they are part of the same task.

3. Modularity

Process cooperation is very important if the system is to be constructed in modular style. The modular system divides the functions into separate processes or threads.

4. Convenience

Process cooperation provides convenience if single user is working on multiple tasks at the same time. It ensures that any conflicts are avoided in the system if the user is using the same data in multiple tasks.

There are two fundamental models of inter-process communication:

- Shared Memory Model
- Message Passing Model

2.7.3 Shared Memory Model

The **shared memory model** establishes a region of memory that is shared by cooperating processes. The processes exchange information by reading and writing data to the shared region. This model is faster than message passing model and provides maximum speed and convenience.

Typically, the shared memory region resides in the address space of the process that creates this region. The other processes requiring to communicate using that shared memory region must attach it to their address space. The operating system normally prevents one process to access the memory of another process. All processes wishing to communicate using this model must agree to remove this restriction. The processes must also ensure that they do not write to the same location simultaneously.

Producer Consumer Example

Below is a very simple example of two cooperating processes. The problem is called the Producer Consumer problem and it uses two processes called producer and consumer.

Producer Process	It produces information that will be consumed by the consumer.
Consumer Process	It consumes information produced by the producer.

Both processes run concurrently. If the consumer has nothing to consume, it waits.

There are two versions of the producer. In version one, the producer can produce an infinite amount of items. This is called the **Unbounded Buffer Producer Consumer Problem**. In the other version, there is a fixed limit to the buffer size. When the buffer is full, the producer must wait until there is some space in the buffer before it can produce a new item.

Bounded Buffer - Shared Memory Solution

Here is a shared memory solution for Bounded Buffer Producer Consumer problem. Both processes have some shared data that can be accessed and updated. The shared data is as follows:

```
// Shared data:
        int n = 5, item, in, out;
        // number of items in buffer is at most 5
        int buffer[n];
// both the consumer and producer are currently looking at buffer element 0.
        in = 0;
        out = 0;
// Producer Process
        while (true)
        {
                while (in+1 % n == out)
                {
                        // Do nothing
                }
                // produce a random item
                buffer[in] = nextProduction;
                in = in + 1 % n;
        }
// Consumer process
        while (true)
        {
                while (in == out)
                {
                        // Do nothing
                }
                nextConsumed = buffer[out];
                out = out + 1 % n;
        }
```

The producer basically just checks to see if there is any space in which to put a newly produced item (outer while loop). If there is no space, then it waits until there is some space. The consumer waits while the buffer is empty. If there is something, it grabs the next item and consumes it. One drawback with this solution is that there is one element of the buffer that is wasted.

2.7.4 Message Passing Model

In **message passing model**, the cooperating processes communicate by exchanging messages with one another. This model is useful for exchanging smaller amount of data. It can also be implemented more easily than shared memory model. It is typically implemented using system calls and is more time-consuming.

Message passing model is typically useful in distributed environment. The processes may reside on different computers connected by a network. The chat program used on the Internet is an example of message passing system. The users can communicate by exchanging messages to one another. A message passing model provides at least two operations:

- Send (message)
- Receive (message)

Two processes can communicate with each other by sending and receiving messages. This communication can take place by establishing a **communication link** between these processes.

2.7.4.1 Naming

The processes that want to communicate must refer to each other using any method. It can be done using direct or indirect communication.

1. Direction Communication

In **direct communication**, each process that wants to communicate must name the sender or recipient of the communication. There are two schemes of direct communication known as symmetric communication and asymmetric communication. The **symmetric communication** requires that both sender and receiver process must name each other to communicate. In **asymmetric communication**, only the sender process must name the recipient. The recipient process does not need to name the sender process.

Send() and **Receive()** functions in symmetric communication are defined as follows:

Send (P, msg)	It is used to send the message **msg** to the process **P**.
Receive (Q, msg)	It is used to receive the message **msg** by the process **Q**.

Send() and **Receive()** functions in asymmetric communication are defined as follows:

Send (P, msg)	It is used to send the message **msg** to the process **P**.
Receive (id, msg)	It is used to receive message **msg** from any process. The variable **id** refers to the name of the process with which the communication has taken place.

The properties of communication link in direct communication are as follows:

- A link is automatically established between the pair of processes that want to communicate. The processes only need to know the identity of each other.
- A link is established between exactly two processes.
- Only one link is established between one pair of processes.

2. Indirect Communication

In **indirect communication**, the messages are sent and received from **mailboxes** or **ports**. A mailbox is an object used by the processes to store and remove messages. Each mailbox has a unique identification. Two processes can communicate using this method only if they have a shared mailbox. A mailbox can be owned by a process or the operating system. If the mailbox is owned by a process then it disappears when that process is terminated.

The operating system must provide a mechanism for the following:

- Create a new mailbox
- Send and receive messages via mailbox
- Delete a mailbox

Send() and **Receive()** functions in asymmetric communication are defined as follows:

Send (M, msg)	It is used to send the message msg to the mailbox M.
Receive (M, msg)	It is used to receive the message msg from the mailbox M.

The properties of communication link in indirect communication are as follows:

- A link is established between a pair of processes only if both have a shared mailbox.
- A link may be associated with more than two processes.
- Each pair of communicating processes may have a number of links with each link referring to one mailbox

Problems in Mailbox System

Some problems with the mailbox system occur when more than two processes share a mailbox. Suppose that Process 1 sends a message to a mailbox that it shares with Process 2 and Process 3. Process 2 and Process 3 both try to receive the message. Who will get it?

Some solutions to the above problem are:

- To allow a link to be established with at most two processes
- To allow only one process at a time to execute a receive operation
- To allow the system to arbitrarily select the receiver. The sender must be notified who received the message.

2.7.4.2 Synchronization

The communication between different processes takes place through calls to **send()** and **receive()** functions. Different design options to implement each function include message passing by **blocking** or **unblocking**. These are also known as **synchronous** or **asynchronous**.

Blocking Send	The sending process is blocked until the message is received by the receiving process or by the mailbox.
Nonblocking Send	The sending process sends the message and resumes options.
Blocking Receive	The receiver process blocks until a message is available.
Nonblocking Receive	The receiver process retrieves either a valid message or a null.

2.7.4.3 Buffering

The messages exchanged via any type of communication reside in a temporary queue. These queues can be implemented in the following ways:

Zero Capacity	The queue has a maximum length of zero. The link cannot have any messages waiting in it. The sender must wait until the recipient receives the message. The two processes must be synchronized for the transfer of a message. The zero-capacity link is referred to as a message-passing system without buffering.
Bounded Capacity	The queue has a finite length of **n**. It can store n number of messages in it. A message is stored in the queue if it is not full and

	the sender continues the execution without any wait. The sender must block until there is a space available in the queue.
Unbounded Capacity	The queues length is infinite and any number of messages can wait in it. The sender is never blocked. That is why the sender is never delayed.

Review Questions

1. What do you know about process and program?

A process is a program in execution. A program is a passive entity. It specifies the logic of data manipulation and I/O actions. A program does not perform the actions itself. It has to be executed to realize the action. A process is an active entity. It performs the actions specified in the program.

2. What is the difference between process and processor?

A process is a program in execution. It is an an active software entity that has states and attributes. It can hold resources such as memory and CPU. A processor is a hardware resource for executing programs. A process is executed on a processor.

3. What is the state of a process?

State of a process is defined by the current activity of that process.

4. What are different process states and their functions?

Different process states are new, reading, running, waiting and terminated. The new state means that a process that has just been created. It has not been admitted to the pool of executable processes by the operating system. The ready state means that a process is prepared to execute. The running state means that the process is being executed. The waiting state means that the process cannot execute until some event occurs such as the completion of an I/O operation. The terminated state means that the process has been released from the pool of executable processes by the operating system.

5. What is a PCB?

Process control block is a data structure. It contains all information for a process to be used for controlling its execution. It also contains the information related to the resource allocated to the process. Every process has its own PCB. The process control block is very important. It should be stored in protected area where the user cannot access it.

6. How many processes can be in running state in a single-processor system?

In a single-processor system, there will never be more than one running process. If there is more, the rest will have to wait until the CPU is free and can be rescheduled.

7. Where do processes wait?

The processes wait in different operating system queues.

8. List some of the queues on a typical system.

Ready, I/O device, job.

9. How are the queues typically implemented?

FIFO queues, trees, linked-lists.

10. How many device queues are there on a system?

One for each device.

11. How does the operating system regain the control of the processor after it gives the control to a user process?

It can regain the control if a timer expires, the process issues a system call or if the process yields.

12. What does the long-term scheduler do?

It determines which jobs belong to the current mix of running/waiting jobs.

13. What does the short-term scheduler do?

It determines which of the current jobs should run in the next CPU burst.

14. Which scheduler must work very fast in order not to waste significant CPU time? Which can be slow?

Fast: short-term. Slow: long-term.

15. What is the degree of multiprogramming?

The number of jobs in the current mix of running/waiting jobs is called the degree of multiprogramming. If the degree of multiprogramming is stable, it means that the number of jobs entering the system is equal to the number of jobs leaving the system.

16. When is the long-term scheduler invoked?

The long-term scheduler is invoked when a job is completed or when the degree of multiprogramming has not been reached yet.

17. What is swapping and what is its purpose?

Swapping is a process of moving a part or all of a process from main memory to disk. The operating system swaps one of the blocked processes to the disk in a suspend queue when no process in main memory is in Ready state. It allows to bring another process into main memory to execute.

18. Why swapping is advantageous?

It can be advantageous to remove processes from memory and thus to reduce the degree of multiprogramming. It can also be necessary to improve the process mix.

19. What is the context switching?

Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as a context switch.

20. What actions are taken by kernel for context-switching between processes?

The operating system generally saves the state of the currently running process. It restores the state of the process to be executed next. The saving the state of a process includes the values of all CPU registers and memory allocations.

21. Is context switching time independent of hardware?

No, the context switching depends on hardware support.

22. Explain hardware mechanism for supporting context switching.

Some processors provide multiple sets of registers. Context switch just changes the pointer to the current register set. If there are more processes than register sets, the system must copy register data to/from memory as before.

23. What is the most serious problem in context switching?

The time (pure overhead) needed to switch from one job to another.

24. What cause processes to be created?

1. System initialization
2. Execution of a process creation system call by a running process
3. A user request to create a new process (like clicking an icon in the GUI)

4. Initiation of a batch file

25. Describe five implementations of the create-new-process mechanism.

a. Parent continues executing.

b. Parent stops executing until children are done.

c. Parent and children share all variables.

d. Children share only a subset of parent's variables.

e. Parents and children share no common resources.

26. What are the two options for process creation in terms of who initiates the creation?

1. Users create processes
2. Processes create processes

27. What are the two options for process creation in terms of execution?

1. A parent process stops until a child process finishes.
2. A parent and child processes run simultaneously.

28. What are the two options for process creation in terms of memory space?

1. Parent & child processes run in different address spaces (Child has a program loaded into it).
2. Parent and child processes share the address space (Child is a duplicate of parent).

29. How parent and child share the address space (memory) after a fork?

When a new process is created using fork() , memory is not requested immediately. The memory for parent process is used jointly by both processes. If new process subsequently accesses part of the memory in write mode, this section is copied before being modified. It is called copy-on-write.

30. What is the function of exec system call?

This system call loads a binary file into memory destroying the memory image of the program containing exec() system call and starts its execution.

31. What is cascading termination?

Operating system does not allow child process to continue if its parent process terminates. All children processes are terminated if a parent process terminates. This is called cascading termination.

32. Describe the producer/consumer problem.

The consumer can't be allowed to use a result until the producer has created that result. The producer cannot be allowed to create results if the buffers are all full.

33. Give examples of producer/consumer pairs.

Compiler/linker, linker/loader, card-reader/line-printer

34. How do you implement a circular array?

Using mod arithmetic, like in clocks.

35. How do you determine if the buffers (or arrays) are all full in circular arrays?

If the input-pointer is one less than the output-pointer in mod arithmetic.

36. How do you determine if the buffers (or arrays) are all empty?

If the input-pointer equals output-pointer.

37. What are advantages of multiprogramming over non-multiprogramming systems?

When a process is blocked, multiprogramming system can run another process in the meantime. Non-multiprogramming system cannot. The first is more efficient. It makes better use of CPU.

38. Can a process make transition from Ready state to Blocked state? Why or why not?

No, it cannot. A process can become blocked only when it issues a request for a resource that is not available. It can make such a request only when it is executing.

39. What is the meaning and purpose of swapping out a process by the medium-term scheduler?

The medium-term scheduler can remove a process from the ready queue and save it to disk thus freeing up its memory and other resources. The process is said to be "swapped out". It may be required because the process requires more resources than are currently available and so needs to wait until the resources are available or because the process mix needs to be altered.

40. Why should operating system provide mechanisms to facilitate process cooperation?

Information Sharing - Several users may which to share the same information e.g. a shared file. The O/S needs to provide a way of allowing concurrent access.

Computation Speedup - Some problems can be solved quicker by sub-dividing it into smaller tasks that can be executed in parallel on several processors.

Modularity - The solution of a problem is structured into parts with well-defined interfaces, and where the parts run in parallel.

Convenience - A user may be running multiple processes to achieve a single goal, or where a utility may invoke multiple components, which interconnect via a pipe structure that attaches the stdout of one stage to stdin of the next etc.

41. What is preemptive multitasking?

Preemptive Multitasking are the process management schemes that include the possibility of removing a process from the CPU even if there is no I/O-request. In this case the process is set ready and has to queue again for getting calculation time. It is said to have been preempted.

42. What is non-preemptive multitasking?

Non-preemptive Multitasking are the process management schemes that do not allow removing a process from CPU until it is finished or issues an I/O-request. With these schemes, it is impossible to set a process directly from running to ready.

43. Which requirements does CPU scheduler satisfy to be seen as performing its task effectively?

1. Efficient use of resources
2. Fair allocation of resources between processes
3. Satisfactory response time to interactive processes

3

THREADS

Chapter Overview

3.1 Threads
    3.1.1 Benefits of Threads
    3.1.2 Difference between Processes and Threads
3.2 Multithreading
    3.2.1 Advantages of Multithreading
3.3 Types of Threads
    3.3.1 Kernel-Level Threads
    3.3.2 User-Level Threads
3.4 Multithreading Models
    3.4.1 Many-to-One Model
    3.4.2 One-to-one Model
    3.4.3 Man-to-Many Model
3.5 Thread Libraries
3.6 Operating-System Examples
Review Questions

3.1 Threads

A thread is a basic unit of CPU utilization. It is also called a **lightweight process**. It is a sequence of instructions within a process. A thread behaves like a process within a process but it does not have its own PCB. Usually, multiple threads are created within a process. The multiple threads in a process allow multiple executions. It can be used to improve application through parallelism. Thread consists of the following:

- Thread ID
- Program Counter
- Register Set
- Stack

A thread shares the following with its peer threads in a particular task:

- Code Section
- Data Section
- Any operating system resources, which are available to the task.

3.1.1 Benefits of Threads

Some important benefits of threads are as follows:

- Threads share common data and do not need to use inter-process communication.
- Threads can take advantage of multiprocessors.
- There is a higher throughput when multiple threads cooperate in a single job. The whole process does not stop if one thread must wait.
- Threads are cheap to create as they only need a stack and storage for registers.
- Threads use very little resources of an operating system. They do not need new address space, global data, program code or operating system resources.
- Context switching becomes fast when working with threads.

3.1.2 Difference between Processes and Threads

The difference between processes and threads is as follows:

Threads	Processes
• A thread is a lightweight entity.	• A process is a heavy weight entity.
• A thread cannot exist without a process.	• There must be at least one thread in process.
• A thread has no data segment or heap.	• A process has code, data and heap segments.
• If a thread ends, the process may still run.	• If a process ends, all threads in it also end.
• The communication among threads occurs via memory.	• The communication among processes occurs via operating system and data copying.
• The creation of threads and context switching is inexpensive.	• The creation of processes and context switching is expensive.

3.2 Multithreading

Multithreading allows operating system to execute different threads simultaneously. Many software packages for modern computer systems are multithreaded. An application is typically implemented as separate process with several threads. For example, a web browser can use one thread to display images and a second thread to retrieve data from the network.

A single application may need to perform many similar tasks in some situations. For example, a web server receives requests for web pages, images and sounds etc. Many clients may be accessing the server at the same time. The web server can service only one client at a time if it runs as a single-threaded process. The clients may have to wait for very long time for the service of the requests.

A solution of this problem is that the server should run a process to accept requests. The server then creates a separate process to service that request. This method was used before the popularity of threads. However, the process creation is very time consuming. It is more efficient to use one process with multiple threads. A multithreaded server can create a separate thread for client requests and another thread to service that request.

Threads are also very important in **remote procedure call (RPC)** system. RPC servers are typically multithreaded. A server services the received message using a separate thread. It allows the server to service several concurrent requests. Many operating system kernels are also multithreaded. Many threads operate in the kernel and each thread performs a specific task such as managing devices or handling interrupts. For example, Solaris creates a set of threads in the kernel for handling interrupts. Linux uses a kernel thread to manage the amount of free memory in the system.

3.2.1 Advantages of Multithreading

Some advantages of using multithreading are as follows:

1. Responsiveness

Multithreading approach increases responsiveness of the process. A process consists of more than one thread. If one thread is blocked or busy in a lengthy calculation, some other thread may still be executing. So the user gets more response from the executing process. For example, a browser allows a user to interact with it while a file is being downloaded.

2. Resource Sharing

All threads of one process share the memory and resources of that process. Secondly, it allows an application to have several different threads within the same address space.

3. Economy

Allocation of memory and resources for process creation is costly. All threads in a process share the resources of that process so it is more economical to create and context switch the threads.

4. Utilization of Multiprocessor Architecture

Multiprocessor architecture allows the facility of **parallel processing.** It is most efficient way of processing. A single threaded process can be executed on one CPU even if there are more processors. Multithreading on a multiprocessor system increases concurrency. Different parts of a multi-threaded process can be executed simultaneously on different processors.

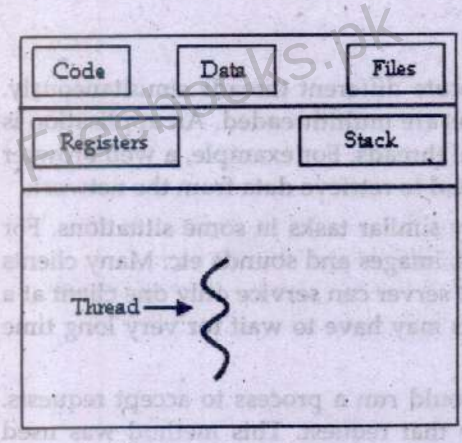


Figure: Single-threaded process

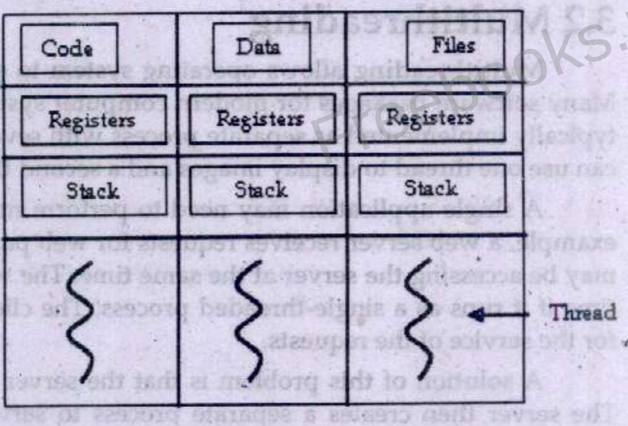


Figure: A multithreaded process

3.3 Types of Threads

Threads may be handled at different levels. There are two types of threads to be managed in a modern system. These are known as **user threads and kernel threads.**

3.3.1 Kernel-Level Threads

Kernel level threads are supported within the kernel of the operating system. All modern operating systems support kernel level threads. They allow the kernel to perform multiple tasks and to service multiple kernel system calls simultaneously.

Advantages

Different advantages of kernel-level threads are as follows:

- Kernel has full knowledge of all threads. Therefore the scheduler may decide to give more time to a process with large number of threads than the process having small number of threads.
- Kernel-level threads are especially good for applications that are blocked frequently.

Disadvantages

Different disadvantages of kernel-level threads are as follows:

- The kernel-level threads are slow and inefficient.
- Kernel must manage and schedule threads as well as the processes. It requires a full thread control block (TCB) for each thread to maintain information about threads. There is significant overhead and increases the kernel complexity.

3.3.2 User-Level Threads

User-level threads are implemented in user-level libraries instead of systems calls. The thread switching does not need to call operating system. It does not cause interrupt to the kernel. The kernel knows nothing about user-level threads. It manages these threads as single-threaded processes. The user-level threads are very fast.

Advantages

Different advantages of user-level threads are as follows:

- The user-level threads are fast and efficient.

- They can be implemented on an operating system that does not support threads.
- They are simple to manage. The switching between threads can be done without kernel.

Disadvantages

Different disadvantages of user-level threads are as follows:

- There is a lack of coordination between threads and operating system kernel. The process gets single time slice even if it has a large number of threads in it.
- A multithreaded application with pure user-level threads cannot take advantage of multiprocessing. The kernel assigns only one process to a processor at a time.

3.4 Multithreading Models

In a specific implementation, the user threads must be mapped to kernel threads using one of the following strategies.

- Many-to-one model
- One-to-one model
- Many-to-many model.

3.4.1 Many-to-One Model

In many-to-one model, many user level threads are mapped to one kernel thread. It is efficient because it is implemented in user space. A process using this model will be blocked entirely if a thread makes a blocking system call. Only one thread can access the kernel at a time so it cannot run in parallel on multiprocessor. The Green threads for Solaris and GNU Portable Threads are implemented by the many-to-one model.

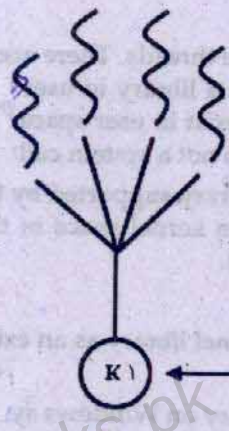


Figure: Many-to-One Model

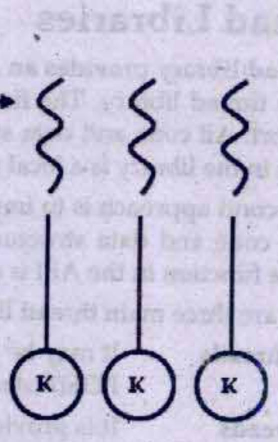


Figure: One-to-one Model

3.4.2 One-to-one Model

In one-to-one model, each user thread is mapped to a kernel thread. It provides more concurrency because it allows another thread to execute if a thread makes a blocking system call. It facilitates parallelism in multiprocessor system. Each user thread requires a kernel thread that may affect the performance of the system. The creation of threads in this model is restricted to a certain number. It is used by Linux and Windows XP and OS/2.

3.4.3 Man-to-Many Model

This model multiplexes many user levei threads to a smaller or equal number of kernel threads. The number of kernel threads may be specific to either a particular application or a particular machine. The user can create any number of user threads and corresponding kernel threads can run in parallel on multiprocessor system. The kernel can execute another thread if a thread makes a blocking system call. The individual processes may be allocated variable numbers of kernel threads depending on the number of CPUs available and other factors. It is used by Solaris, IRIX, HP-UX and Tru64 UNIX.

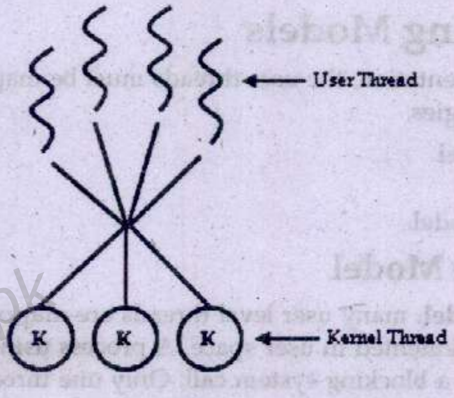


Figure: Many-to-Many Model

3.5 Thread Libraries

A thread library provides an API to create and manage threads. There are two ways to implement a thread library. The first approach is to provide a library in user space with no kernel support. All code and data structures for the library exist in user space. The invoking of a function in the library is a local function call in user space not a system call.

The second approach is to implement a kernel-level library supported by the operating system. The code and data structures for the library exist in kernel space in this case. The invoking of a function in the API is a system call to the kernel.

There are three main thread libraries in use today:

1. POSIX Pthreads	It may be provided as a user or kernel library as an extension to the POSIX standard.
2. Win32 threads	It is provided as a kernel-level library on Windows systems.
3. Java threads	Java generally runs on a Java Virtual Machine. The implementation of threads is based on the operating system and the hardware on which JVM runs.

3.6 Operating-System Examples

Some examples of operating system examples of threads are as follows:

1. Windows XP Threads

Win32 API thread library supports one-to-one thread model. Win32 also provides fiber library that supports many-to-many model.

Win32 thread components include the following:

- Thread ID
- Registers
- A user stack used in user mode and a kernel stack used in kernel mode
- A private storage area used by various run-time libraries and dynamic link libraries.

The key data structures for Windows threads are **ETHREAD (executive thread block), KTHREAD (kernel thread block) and TEB (thread environment block)**. The ETHREAD and KTHREAD structures exist entirely within kernel space and can only be accessed by kernel. The TEB lies within the user space.

2. Linux Threads

Linux does not distinguish between processes and threads. It uses the more generic term tasks. The traditional **fork()** system call completely duplicates a process task. An alternative system call **clone( )** is used for varying degrees of sharing between the parent and child tasks. It is controlled by different flags.

Review Questions

1. What are threads?

A thread is a sequence of instructions within a process. It is the basic unit of CPU allocation in modern operating systems. It has its own stack and program counter. However, it does not have its own PCB. Usually multiple threads are created within a process.

2. Which attributes are used by operating system to keep track of the information about a thread?

The attributes used by operating system to keep track of the information about a thread include program counter (PC), registers, stack and thread ID.

3. List key differences between processes and threads.

A process is heavyweight but a thread is lightweight. Thread is part of a process and process must have at least one thread. A process can have multiple threads. Threads provides better control for tasks and scheduling etc. IPC and sharing is easier between threads than processes.

4. List some similarities between processes and threads.

Both process and thread are supported on most modern operating systems. They have many attributes such as owner and priority etc. Both have stack space and program counter. Both require CPU scheduling in order to do their work.

5. In what ways are threads better than processes?

Threads share some of the same resources as other threads. CPU does not need to save/restore as many registers during a context switch.

6. What is multithreading?

Multithreading is a technique in which a process executing an application is divided into threads that can run concurrently.

7. Compare two different threads running in the same process and two independent processes. What is preferable at which time?

Threads running in the same process share all processes resources like address space, code, files, etc. Different processes typically do not share any of these things. They can share files, code and memory. Threads in the same process are useful when it is required to execute two different but very closely related pieces of code that must share a lot of data, state, etc. Different processes are more appropriate when you want to execute two different and unrelated pieces of code that do not share a significant amount of data or state.

8. Compare cooperative and preemptive multithreading.

In cooperative multithreading, the running thread must explicitly yield the CPU so that the dispatcher can select a ready thread to run next. In preemptive multithreading, an external event such as interrupt invokes the dispatcher to preempt the running thread and select a ready thread to run next.

Cooperative multithreading relies on the cooperation of threads to ensure each thread receives regular CPU time. Preemptive multithreading enforces a regular allocation of CPU time to each thread even when a thread is uncooperative.

9. Describe user-level threads and kernel-level threads.

User-level threads are implemented in the application. The thread management structures (Thread Control Blocks) and scheduler are contained within the application. The kernel has no knowledge of the user-level threads. Kernel threads are implemented in the kernel. The TCBs are managed by the kernel. The thread scheduler is the normal in-kernel scheduler.

10. List the advantages of user level thread over kernel level thread.

Thread switching does not require kernel mode privileges because all thread management data structures are in the user address space of a single process. The process does not switch to the kernel mode to do thread management. The scheduling algorithm can be applied to the application without disturbing the operating system scheduler. User level threads can run on any operating system. No changes are required to the kernel to support user level threads.

11. List two disadvantages of user level thread to kernel level thread.

Many system calls are blocking in a typical operating system. When a user level thread executes a system call, it blocks that thread as well as all of the threads within the process. In a pure user level thread strategy, multithreaded application cannot take advantage of multiprocessing. A kernel assigns one process to only one processor at a time

12. What is a thread of execution in a process?

A thread of execution has a program counter that keeps track of which instruction to execute next. It has registers that holds it's current working variables. It has a stack that contains the execution history with one address for each procedure called but not yet returned from.

13. Differentiate between 3 processes each with 1 thread and 1 process with 3 threads?

In first case, each thread operates in a different address space. They are essentially unrelated. In the second case, all three threads share the same address space. They are actually part of the same job and are actively and closely cooperating with each other.

14. What is a blocked thread? Give an example.

A blocked thread is waiting for some event to unblock it. For example when a thread performs a system call to read from the keyboard, it is blocked until input is typed.

15. Why does each thread need its own stack?

Each thread generally calls different procedures and a different execution history. Suppose procedure X of thread1 calls procedure Y and procedure Y calls procedure Z. While Z is executing, the return addresses for X, Y and Z will be on the stack. At the same time thread2 may have completely different history. This is why is thread need its own stack.

16. What multiple threads add to the process model?

Allows multiple executions to take place in the same process environment to a large degree independent of one another.

17. Is there any protection between threads of a process?

There is no protection between threads of a process. Every thread can access every memory address within the process' address space. One thread can read, write or delete another thread's stack.

18. What is the main advantage of multi-threaded model for programmer?

The model allows the server to be written as a collection of sequential threads.

19. How threads are more efficient in multiprocessor system?

The benefits of multithreading can be greatly increased in a multiprocessor architecture. Each thread may be running in parallel on a different processor.

20. What is the benefit of multiprocessor system for a single-threaded process?

A single-threaded process can only run on one CPU even if are available. As CPU can run several processes at any time, requests for the CPU will be served quickly. CPU will be accessible when it is needed by single-threaded process so, shorter waiting time and better performance.

21. How does multi-threaded process work in a single-processor system?

CPU generally moves between each thread quickly to create an illusion of parallelism. In reality one thread is running at a time. Only multi-threading on a multi-CPU system increases concurrency.

22. How a new thread is created?

The process normally starts with a single thread when multithreading is present. This thread has the ability to create new threads by calling a library procedure.

4

CPU SCHEDULING

Chapter Overview

4.1 CPU Scheduling
4.2 Multiprogramming
4.3 Scheduling Objectives
4.4 CPU I/O Burst Cycle
4.5 Working of CPU Scheduler
4.6 Preemptive and Non-Preemptive Scheduling
4.7 Interval Timer
4.8 Dispatcher
4.9 Scheduling Criteria
4.10 Scheduling Algorithms
    4.10.1 First Come First Served Scheduling
    4.10.2 Shortest Job First Scheduling
        4.10.2.1 Non-Preemptive SJF Example
        4.10.2.2 Preemptive SJF example
    4.10.3 Priority Scheduling
        4.10.3.1 Non-Preemptive Example
        4.10.3.2 Preemptive Example
    4.10.4 Round Robin Scheduling
    4.10.5 Multi-Level Queue Scheduling
    4.10.6 Multi-Level Feedback Queue Scheduling
4.11 Multiple Processor Scheduling
4.12 Real-Time Scheduling
4.13 Algorithm Evaluation
    4.13.1 Deterministic Modeling
    4.13.2 Queuing Models
    4.13.3 Simulation
    4.13.4 Implementation

Review Questions

4.1 CPU Scheduling

A major task of an operating system is to manage a collection of processes. In some cases, a single process may consist of a set of individual threads. In both situations, a system with a single CPU or a multi-processor system with fewer CPU's than processes has to divide CPU time among the different processes or threads that are competing to use it. This process is called **CPU scheduling.**

4.2 Multiprogramming

The operating system can make the computer more productive by switching the CPU among processes. The objective of multiprogramming is to have some process running at all times. This should maximize CPU utilization. If there is only one processor (or CPU) then there will only be one process in a running state at any given time. If there are more processes that need to be run, they will have to wait. There must be some mechanism to select which process (currently in memory) will run next when the CPU becomes available.

The operating system maintains a **ready-queue**. Processes on this queue are ready to be executed. Whenever a currently executing process needs to wait (does I/O etc.), operating system picks a process from the ready queue and assigns the CPU to that process.

Scheduling is a fundamental operating system function. CPU is an important resource. It is very important to develop good scheduling algorithms.

Suppose we have two processes A and B to be executed as shown in the figure. Each process executes for one second then waits for one second. It is repeated 60 times. If we run process A first and then process B, one after the other, it will take four minutes to run the two processes: A takes two minutes to run and then B takes two minutes to run. We actually compute only two minutes of this time. The other two minutes are idle time. Our CPU utilization is only 50%.

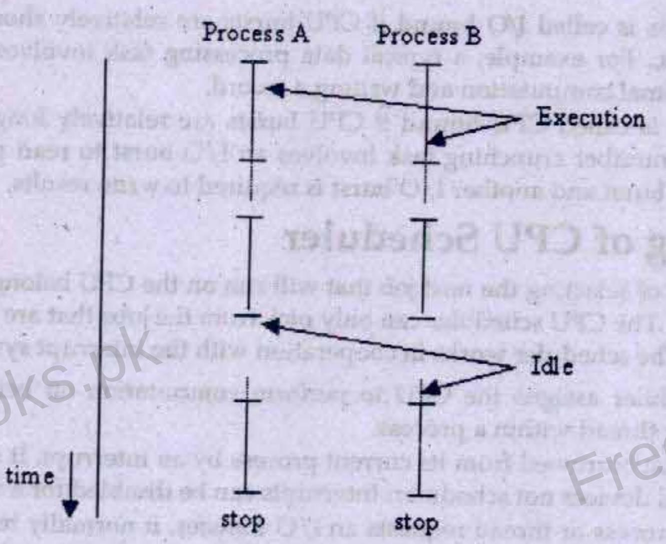


Figure: Two processes A and B for execution

4.3 Scheduling Objectives

Schedulers typically attempt to achieve some combination of the following goals. These goals are contradictory to some extent:

- Maximize CPU utilization (due to its relatively high cost)
- Maximize utilization of other resources (disks, printers etc.)
- Maximize **throughput** = number of jobs completed per unit time.
- Minimize **waiting time** = total time a job spends waiting in the various queues for a resource to become available.
- Minimize **turnaround time** = waiting time + computation time + I/O time
- Minimize **response time** (timesharing) = time from entry of a command until first output starts to appear.
- Fairness: All comparable jobs should be treated equally.
- Avoid indefinite postponement. It is normally avoided by using **aging**. In aging, the priority of a process grows as it waits for a resource. The priority slowly becomes highest and it gets the resource.
- Uniformity: The behavior of the system should be predictable.
- Graceful degradation: The system response deteriorates gradually rather than coming to a sudden virtual standstill in case of excessive loads.
- Predictability: A given job should run in about the same amount of time and at about same cost regardless of system load.

4.4 CPU I/O Burst Cycle

The execution of a process consists of an alternation of **CPU bursts and I/O bursts**. A process begins and ends with a CPU burst. In between, CPU activity is suspended whenever an I/O operation is needed.

- The process is called **I/O bound** if CPU bursts are relatively short compared to the I/O bursts,. For example, a typical data processing task involves reading a record, some minimal computation and writing a record.
- A process is called **CPU bound** if CPU bursts are relatively long compared to I/O bursts. A number crunching task involves an I/O burst to read parameters. A very long CPU burst and another I/O burst is required to write results.

4.5 Working of CPU Scheduler

The process of selecting the next job that will run on the CPU belongs to the short-term or CPU scheduler. The CPU scheduler can only pick from the jobs that are already in memory and ready to go. The scheduler works in cooperation with the interrupt system.

- The scheduler assigns the CPU to perform computation on behalf of a particular process or thread within a process.
- CPU can be borrowed from its current process by an interrupt. It is under the control of external devices not scheduler. Interrupts can be disabled for a short time.
- When a process or thread requests an I/O transfer, it normally becomes ineligible to use the CPU until the transfer is complete. This means that the scheduler will have to choose a new process or a new thread within the same process to use the CPU.
- The process or thread that requested the I/O again becomes eligible to use the CPU when the I/O transfer is complete.

4.6 Preemptive and Non-Preemptive Scheduling

There are four conditions under which CPU scheduling may take place. They are:

1. When a process switches from the running state to the waiting state
2. When a process switches from the running state to the ready state
3. When a process switches from the waiting state to the ready state
4. When a process terminates

If only conditions 1 and 4 apply, the scheduling is called **non-preemptive.** All other scheduling is **preemptive.** Preemption means that a process may be forcibly removed from the CPU even if it does not want to release the CPU. It is because a higher priority process needs the CPU.

Once a process gets CPU, the simplest approach is to allow the process to continue using CPU until it voluntarily yields CPU by requesting an I/O transfer etc. I/O interrupts may steal CPU from time to time. After each interrupt, control passes back to the process that was running when it occured. This is called a **non-preemptive** approach.

In a preemptive scheme, a running process may be forced to yield CPU by an external event instead of its own action. Such external events can be any or both of the following:

- A higher priority process enters the system from outside.
- A higher-priority process that was in the wait state becomes ready. This could occur as the result of an I/O interrupt that moves a waiting process to the ready list.

4.7 Interval Timer

Timer interruption is a technique that is closely related to preemption. When a process gets the CPU, a timer may be set to a specified interval. If the process is still using the CPU at the end of the interval, then it is preempted. Both timer interruption and preemption force a process to yield the CPU before its CPU burst is complete.

However, it is helpful to distinguish timer interruption from preemption caused by higher priority processes becoming ready for two reasons:

- Timer interruption is a function of the particular process's own behavior. It is independent of the rest of the system.
- Almost all multi-programmed operating systems use some form of timer to prevent a process from tying up the system forever. But preemption for a higher priority process is a feature that may or may not be included in a given operating system.

4.8 Dispatcher

Dispatcher is a program that actually gives control of CPU to a process selected by CPU scheduler. It is another part of the scheduling system.

The functions of dispatcher module are as follows:

- Switching context
- Switching to user mode
- Jumping to the proper location in the user program to restart it

The dispatcher should be very fast because it is called every time a process takes control of CPU. The time that the dispatcher takes between stopping one process and starting another process is called the **dispatch latency.**

4.9 Scheduling Criteria

There are many scheduling algorithms and various criteria to judge their performance. Different algorithms may favor different types of processes. Some criteria are as follows:

CPU utilization	CPU must be as busy as possible in performing different tasks. CPU utilization is more important in real-time system and multi-programmed systems.
Throughput	The number of processes executed in a specified time period is called throughput. The throughput increases for short processes. It decreases if the size of processes is huge.
Turnaround Time	The amount of time that is needed to execute a process is called turnaround time. It is the actual job time plus the waiting time.
Waiting Time	The amount of time the process has waited is called waiting time. It is the turnaround time minus actual job time.
Response Time	The amount of time between a request is submitted and the first response is produced is called response time.

A CPU scheduling algorithm should try to maximize the following:

- CPU utilization
- Throughput

A CPU scheduling algorithm should try to minimize the following:

- Turnaround time
- Waiting time
- Response time

4.10 Scheduling Algorithms

Below is a list of some well known scheduling algorithms:

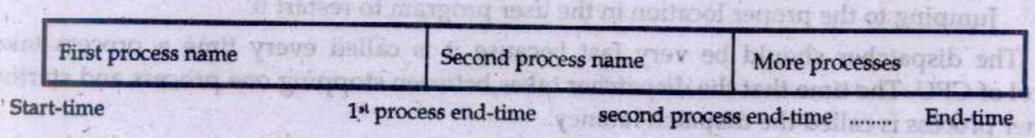
- First Come First Served (FCFS) Scheduling - non-preemptive
- Shortest Job First (SJF) Scheduling - preemptive or non-preemptive
- Priority Scheduling - preemptive or non-preemptive
- Round Robin Scheduling - preemptive

Each scheduling algorithm has its own criteria to choose the next job that will run on CPU. Since CPU scheduler needs to be fast, actual algorithms are typically not very complex.

Timelines

Scheduling is based on the information that is available at a given time. We need some way to represent the state of the system and any processes in it and how it changes over time. Gantt charts are used for this purpose. A Gantt chart is basically a glorified timeline that is used to depict the order of process execution graphically. Below is an outline of a Gantt chart:

Gantt chart



4.10.1 First Come First Served Scheduling

First Come First Served is the simplest CPU scheduling algorithm. It says that the process that enters first should get. FCFS algorithm is easy to understand. It is basically a queue like those we see in banks, shops, etc.

A drawback of the FCFS algorithm is that the processes may have to wait for excessively long amounts of time.

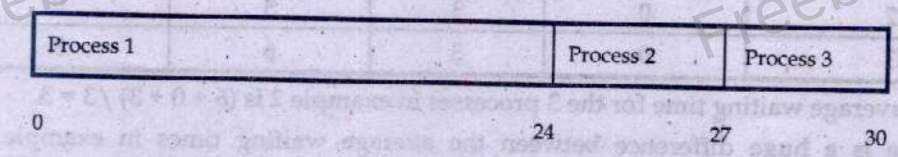
Example 1

Suppose that there are three processes that arrive in the order shown below. They all arrive at time 0, but the system has decided to serve them in this order.

Process name	Start time	Burst time
Process 1	0	24
Process 2	0	3
Process 3	0	3

Gantt Chart for FCFS Example 1

If the processes are served in FCFS order, the Gantt chart would look like as follows:



Process 1 is dispatched first and runs for 24 time units (there is no preemption). Once process 1 has finished, process 2 is dispatched and runs for 3 time units. 24 time units have already elapsed, so process 2 starts at time 24. Once process 2 has finished at time 27, process 3 is dispatched. When process 3 has finished there have been 30 time units used (24 + 3 + 3).

The waiting time for a particular process can be measured by taking its finish time and subtracting the burst time and its arrival time.

The following table shows the waiting time for each process:

Process name	Arrival time	Burst time	Finish time	Waiting time
Process 1	0	24	24	0
Process 2	0	3	27	24
Process 3	0	3	30	27

Waiting time = Finish time - burst time - arrival time

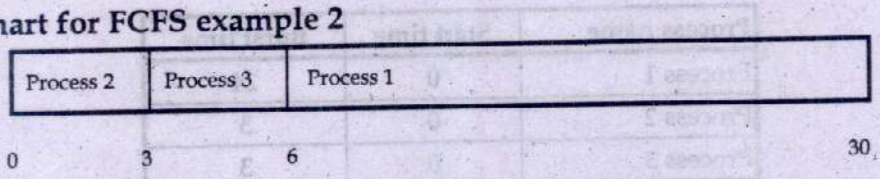
The average waiting time for a whole set of processes can be calculated by adding all the individual waiting times and dividing by the number of processes. In the above example, the average waiting time is: (0 + 24 + 27) / 3 = 17.

Example 2

The waiting time is obviously dependent on the order in which the processes are served. Since it was mentioned that all three processes arrive at time 0, the system could have chosen to dispatch them in the following order:

Process name	Arrival time	Burst time
Process 2	0	3
Process 3	0	3
Process 1	0	24

Gantt chart for FCFS example 2



The waiting times for the processes in example 2 are:

Process name	Arrival time	Burst time	Finish time	Waiting time
Process 1	0	24	30	6
Process 2	0	3	3	0
Process 3	0	3	6	3

The average waiting time for the 3 processes in example 2 is (6 + 0 + 3) /3 = 3.

There is a huge difference between the average waiting times in example 1 and example 2, even though the same three processes are being executed. Since FCFS is not preemptive, it is not a good scheduling algorithm for systems where response time is important (i.e. real time or time sharing systems).

4.10.2 Shortest Job First Scheduling

Another way to schedule jobs is to pick the job that will take the least amount of time to complete. In FCFS scheduling, the average waiting time could be reduced by running the short jobs first. The SJF scheduling algorithm picks the shortest job in terms of burst size and places it on the CPU.

There are two possible schemes of this algorithm:

1. Non-preemptive	Once the CPU is given a process it cannot be preempted until the current CPU burst finishes.
2. Preemptive	If a new process arrives with a shorter CPU burst than remaining CPU burst of the currently executing process, it replaces currently executing process. It is also called **Shortest Remaining Time First**.

Shortest job first scheduling gives the minimum average waiting time for a given set of processes. If you run all the short jobs first, then each subsequent job has a relatively short waiting time. If you were to run all the long jobs first, then every subsequent job would have a longer time to wait.

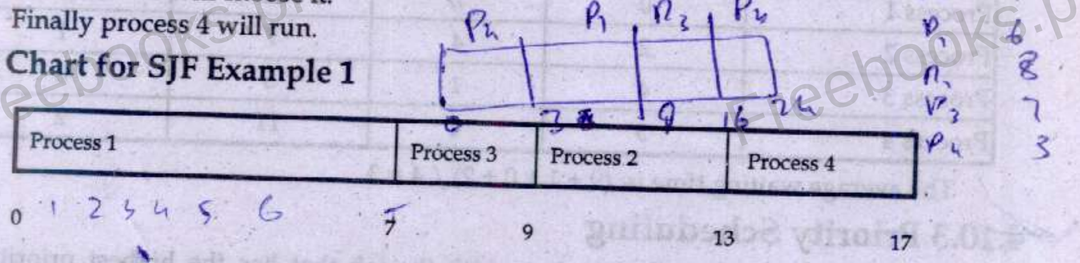
4.10.2.1 Non-Preemptive SJF Example

In the following example there are four processes.

Process name	Arrival time	Burst time
Process 1	0	7
Process 2	2	4
Process 3	4	2
Process 4	5	4

- The arrival times in this example are important. Since there is only 1 process available at time 0 i.e. Process 1, it is by default the shortest process.
- At time 2, process 2 arrives but process 1 is already running. There is no preemption, so even though process 2 is now the shortest it has to wait until process 1 finishes.
- At time 4, process 3 arrives that is the shortest but process 1 has still not finished.
- At time 5, process 4 arrives and process 3 is still the shortest. When process 1 finishes, then process 3 will run because it is the shortest.
- When process 3 finishes, process 2 will run. It has the same length as 4, but arrived earlier so we will choose it.
- Finally process 4 will run.

Gantt Chart for SJF Example 1



The waiting times for the processes in this example are:

Process name	Arrival time	Burst time	Finish time	Waiting time
Process 1	0	7	7	0
Process 2	2	4	13	7
Process 3	4	2	9	3
Process 4	5	4	17	8

The average waiting time is: (0 + 9 + 3 + 8) / 4 = 5

4.10.2.2 Preemptive SJF example

In the following example there are again four processes. Note that process 3 now has a burst time of 1.

Note

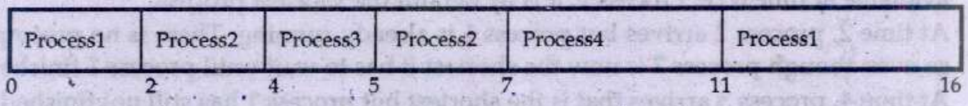
Process name	Arrival time	Burst time
Process 1	0	7 - 2 = 5
Process 2	2	4 - 2
Process 3	4	1 ✓
Process 4	5	4

5 4 –
2
4 –

0
1
2
3
4
5

- The arrival times in this example are important. Since there is only 1 process available at time 0 i.e. Process 1, it is by default the shortest process.
- At time 2, process 2 arrives, but process 1 is already running. However, there is preemption, so even though process 1 is already running, it is removed from the CPU and replaced with process 2.
- At time 4, process 3 arrives that is now the shortest and it replaces process 2 on CPU.
- At time 5, process 3 finishes and process 4 arrives. Process 2 is once again the shortest and it gains the CPU, running until time 7.
- At time 7 process 4 is the shortest and runs until time 11.
- Finally, process 1, which is the only process left, is now the shortest and finishes.

Gantt chart for SJF example 2



The waiting times for the processes in this example are:

Process name	Arrival time	Burst time	Finish time	Waiting time
Process 1	0	7	16	9
Process 2	2	4	7	1
Process 3	4	1	5	0
Process 4	5	4	11	2

The average waiting time is: (9 + 1 + 0 + 2) / 4 = 3

4.10.3 Priority Scheduling

Another way to schedule jobs is to pick the job that has the highest priority. This requires that each process should have a priority associated with it. The priority is generally an integer with some well-defined range e.g. 1 to 10. The CPU is allocated to the process with the highest priority. In some systems, smaller numbers mean higher priority, in other systems larger numbers mean higher priority.

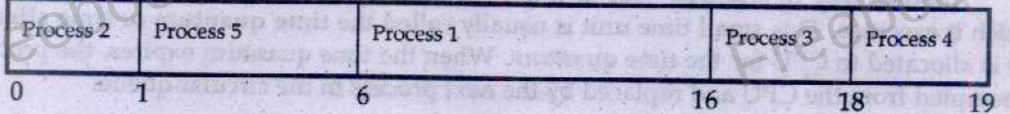
Priority scheduling can either be non-preemptive or preemptive. If there is preemption then a high priority job can remove a low priority job from the CPU and take over.

4.10.3.1 Non-Preemptive Example

Process name	Burst Time	Priority
Process 1	10	3
Process 2	1	1
Process 3	2	3
Process 4	1	4
Process 5	5	2

Assume that all processes arrived at time 0. First of all, Process 2 will get the CPU as it has the highest priority 1 and will finish. Then Process 5 will start execution. In non-preemptive priority scheduling, once a process gets the CPU, it will finish its work and then lease the CPU.

Gantt chart for Non-preemptive Priority Scheduling



The average waiting time is: (1 + 6 + 16 + 18 + 19) / 5 = 8.2

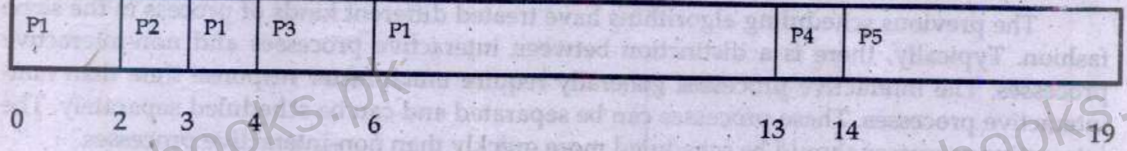
4.10.3.2 Preemptive Example

Process name	Arrival Time	Burst Time	Priority
Process 1	0	10	3
Process 2	2	1	1
Process 3	4	2	2
Process 4	6	1	4
Process 5	8	5	5

In the above example the execution takes place in the following sequence:

- Process 1 arrives at time 0 and starts execution, as there is no other process.
- At time 2, Process 2 arrives with higher priority, so Process 1 will be preempted and Process 2 will get the CPU and run to the completion using one unit of time at time 3.
- At this point, there is again only one Process 1, which resumes and gets the CPU.
- At time 4, Process 3 arrives with higher priority than Process 1 and gets the CPU and completes its execution.
- At time 6, Process 4 arrives with priority 4. Now there are two processes i.e. Process 1 and Process 4. So, Process 1 continues using CPU.
- At time 8, the last process arrives with a priority 5. Process 1 is still a high priority process, so it will run the completion.
- As P1 releases the CPU after completing its job, it will be assigned to P4, which will execute completely.
- In the end, P5 will get the CPU.

Gantt Chart for Preemptive Priority Scheduling



Shortest Job First scheduling is also a form of priority scheduling. In the case of Shortest Job First scheduling, the priority is defined as the predicted next CPU burst.

One major problem with Priority based scheduling is that it may not be fair. Some low priority processes may not ever get the chance to execute because higher priority processes keep stealing the CPU. One solution to this problem is to implement aging. The process of increasing the priority of a process as it gets older is known as aging.

4.10.4 Round Robin Scheduling

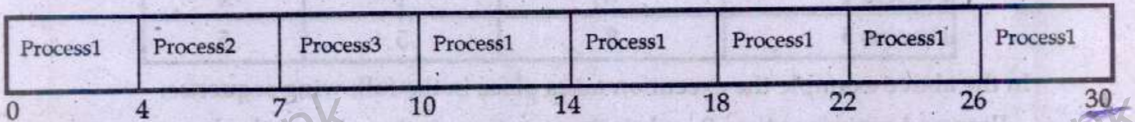
Another way to schedule jobs is to assign a small amount of time to each process in which it executes. This small time unit is usually called the **time quantum or time slice**. The job is allocated to CPU for the time quantum. When the time quantum expires, the process is preempted from the CPU and replaced by the next process in the circular queue.

Example

Process name	Burst Time
Process 1	24
Process 2	3
Process 3	3

Finish | waity
30 | 6 | 30
7 | 4 | 7
10 | 7 | 10
47
avg

Gantt chart for Round-Robin Scheduling



Process1 gets the first 4 units of time. It requires another 20 units so it is preempted after the first time quantum. CPU is given to Process2 that does not require 4 units. It quits before its time quantum ends and CPU is given to Process3. Once each process has got one time quantum, CPUis given to Process1 for additional time quantum. Average turnaround time is 47/3 = 16.

Round Robin algorithm does not allocate CPU to any process for more than one time quantum in a row. If CPU burst exceeds time quantum, it is preempted and placed in the ready queue. That is why Round Robin is called preemptive scheduling algorithm.

The Round Robin scheduling algorithm will be similar to FCFS if the time quantum is very large. If the time quantum is larger than the largest CPU burst for all of the processes, then no process will be preempted. If the time quantum is made too small, then the context switch will take too much time. It is required when preempting one process from the CPU and replacing it with another.

4.10.5 Multi-Level Queue Scheduling

The previous scheduling algorithms have treated different kinds of process in the same fashion. Typically, there is a distinction between interactive processes and non-interactive processes. The interactive processes generally require much faster response time than non-interactive processes. These processes can be separated and can be scheduled separately. The interactive processes should be scheduled more quickly than non-interactive processes.

The ready queue can be split into different sub-queues. One sub-queue can be used for each type of processes. An example of a 5 level queuing strategy is shown below. The 0 has the highest priority and 4 has lowest priority.

0. System processes
1. Interactive processes
2. Interactive editing processes
3. Batch processes
4. Student processes

The type of a process is automatically identified when it enters the system. It is allocated to the correct queue. There is no way to change queues once a process has been allocated to a queue. There are five queues in the above example. The individual jobs as well as five queues need to be scheduled. Two schemes to schedule the queues are as follows:

1. Fixed (Absolute) Priority

In this scheme, jobs are first serviced from queue 1. If there is no job in queue 1, then the jobs in queue 2 are serviced. Similarly, jobs in queue 3 are serviced only if there are no jobs in queues 1 or 2, and so on.

2. Time Slice

In this scheme, each queue gets a time slice. The queues themselves are in a Round Robin queue. Jobs in each queue are executed for the specified amount of time. The control is then transferred to the next queue.

4.10.6 Multi-Level Feedback Queue Scheduling

The main difference between a multi-level queuing strategy and a multi-level feedback queuing strategy is that jobs may move from one queue to another over time. It is possible to implement aging this way. A process moves up to next higher priority queue if it stays in one queue too long. The multi-level feedback queue strategy is defined by following parameters:

- Number of queues
- Scheduling algorithm for each queue
- Method used to determine when to upgrade a process
- Method used to determine when to demote a process
- Method to determine the queue a process will enter when that process needs service

Example

Suppose there are three queues:

- Q_0 - time quantum 8 milliseconds
- Q_1 - time quantum 16 milliseconds
- Q_2 - FCFS

The scheduling will be implemented as follows:

A new job enters queue Q_0, which is served FCFS. When it gains CPU, job receives 8 milliseconds. If it does not finish in 8 milliseconds, job is moved to queue Q_1. At Q_1, job is again served FCFS and receives 16 additional milliseconds. If it still does not complete, it is preempted and moved to queue Q_2.

4.11 Multiple Processor Scheduling

CPU scheduling becomes much more complex when multiple CPUs are available. It is now possible for more than one job to run on a machine at any one time. The number of running jobs is equal to the the number of processors.

An important issue is the access to resources. Suppose only processor A has access to a certain device. Others cannot access that device. Any process that requires that device has to be scheduled on processor A as other processors cannot satisfy its requests.

Another issue is that of contention. If two processes run on separate processors, they may try to update some shared data. It is also possible that two processors may try to grab the same process from the ready queue.

Some systems assign one master processor to do all the scheduling work. The master-processor hands out the work to the slave processors who actually do the work. This ensures that there is only one processor responsible for scheduling work. It should have all the information required to properly perform the scheduling. This approach requires no data sharing between processors.

4.12 Real-Time Scheduling

There are two types of real time systems:

1. Hard real-time systems	The scheduler must ensure that a task is completed within a guaranteed amount of time.
2. Soft real-time systems	The scheduler must ensure that higher priority processes will run first. There is no guarantee that the process will finish within a specified time limit. The only guarantee is that process will finish as soon as possible.

4.13 Algorithm Evaluation

There are a number of scheduling algorithms and strategies. The choice of algorithms depends heavily on the most important criteria. Once an algorithm is selected, it should evaluated. Here are some ways to evaluate an algorithm:

4.13.1 Deterministic Modeling

It takes a particular predetermined workload (set of processes) and defines the performance of the algorithm dependant on that workload. This form of evaluation is simple and can be done quickly. However, it is not very general. We might get a completely different idea about the performance of an algorithm if we chose a different set of processes.

4.13.2 Queuing Models

We can examine the distribution of CPU burst and I/O wait times. We can apply a formula and compute one of three things

- Average length of the queue
- Average waiting time of the queue
- Average arrival rate.

This approach does have some limitations. For example, the results may not be very exact as they rely on an approximation of the actual system.

4.13.3 Simulation

A lot of processes are required to simulate an operating system. The processes should mimic those of typical users and the operating system itself that can be difficult. The simulation may take a long time to complete. For more accurate results, it may be necessary to use process information from a real system.

4.13.4 Implementation

The only real way to test an operating system is to write the code and run it. However, this approach is very expensive.

Review Questions

1. What is a CPU burst and an I/O burst?

CPU burst is a time interval when a process uses CPU only. I/O burst is a time interval when a process uses I/O devices only.

2. What do you mean by CPU-bound and I/O-bound?

CPU-bound means that a job needs more computation and CPU time to finish. I/O-bound means that a job needs more input and output handling from files or users. It spends more time in I/O wait state than in active use of CPU.

3. An I/O-bound program would typically have what kind of CPU burst?

Short.

4. What is FIFO? — FCFS

First-in-first-out queue.

5. What does "preemptive" mean?

Cause one process to temporarily halt, in order to run another.

6. What is the "dispatcher"?

It is a program that gives control of the CPU to the process selected by the short-term scheduler.

7. List performance criteria we could select to optimize our system.

CPU use, throughput, turnaround time, waiting time, response time.

8. What is throughput?

Number of jobs done per time period.

9. What are turnaround time and response time?

Turnaround time is the interval between the submission of a job and its completion. Response time is the interval between submission of a request, and the first response to that request.

10. What is FCFS?

First-Come-First-Serve.

11. What is a Gantt chart? Explain how it is used.

It is a rectangle marked off horizontally in time units, marked off at end of each job or job-segment. It shows the distribution of time-bursts in time. It is used to determine total and average statistics on jobs processed by formulating various scheduling algorithms on it.

12. What is SJF?

It stands for Shortest job first. It usually means the job with the shortest CPU burst.

13. What is indefinite blocking? How can it occur?

It is also called starvation. A process with low priority that never gets a chance to execute can occur if CPU is continually busy with higher priority jobs.

14. What is "aging"?

It is the gradual increase of priority with age of job to prevent "starvation."

15. What is SRTF (Shortest-Remaining-Time-First) scheduling?

A preemptive scheduling algorithm that gives high priority to a job with least amount of CPU burst left to complete.

16. What is round robin scheduling?

Each job is given a time quantum slice to run. If it is not completely done by that time interval, job is suspended and another job is continued. After all other jobs have been given a quantum, first job gets its chance again.

17. True or False: Round robin scheduling is preemptive.

True.

18. What is the time quantum used for?

Time quantum is used in Round robin scheduling to give each process the same processing time.

19. How should the time quantum be related to the context switch time?

Quantum should be very large compared to context switch time.

20. How should the time quantum be related to the CPU burst times?

80% of CPU bursts should be shorter than time quantum.

21. Describe the foreground-background approach.

Low priority processes run in background. High priority jobs run in foreground. Background runs only when foreground is empty or waiting for I/O.

22. How can multilevel queues be scheduled? Which might have priority over others?

a. Each queue can have absolute priority over lower queues.

b. Time-slice queues can, giving each queue a certain percent of time.

23. What are multilevel feedback queues?

Processes move from one queue to another depending on changes in its conditions (the CPU burst may change).

24. Consider the following set of processes to be executed on a system that uses preemptive shortest job first (Shortest remaining time first) scheduling algorithm.

Process	Arrival time	CPU burst (in ms)
P1	0	20
P2	5	16
P3	6	10
P4	8	3
P5	9	1

15 14

1. Show the Gantt chart.

P1	P3	P4	P5	P4	P3	P1	P2
0	6	8	9	10	12	20	34

2. What will be the average waiting time?
   a. 45.7 ms
   b. 9.6 ms
   c. 12.6 ms
   d. The correct answer is ______
   Answer: b

3. If process P1 terminates after its CPU burst given above, then what is the turn around time for process?
   a. 34
   b. 50
   c. 26
   d. The correct answer is ______
   Answer: a

4. What is the waiting time for process P4?
   - a. 1
   - b. 2
   - c. 3
   - d. The correct answer is ______

   Answer: a

25. Consider the following set of processes:

Process	Burst-Time	Priority
P1	15	4
P2	3	8
P3	7	5
P4	5	2
P5	1	5
P6	2	5

The processes are assumed to arrive in the order P1, P2, P3, P4, P5, P6, all at time 0.

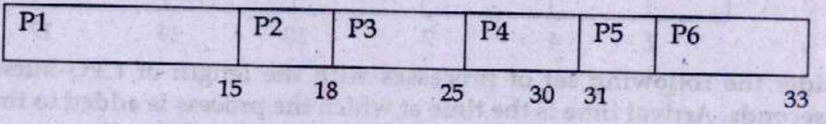
Use FCFS, SJF, Round Robin (quantum2), Priority (low numbers have high priority). For each of the algorithms:

- Draw a Gantt chart
- Obtain the average waiting time
- Obtain the average turnaround time

FCFS:

AWT=(0+15+18+25+30+31)/6=19.83

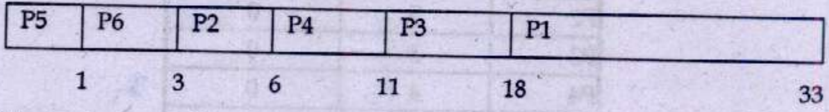
ATT=(15+18+25+30+31+33)/6=25.33



SJF:

AWT=(18+3+11+6+0+1)/6=6.5

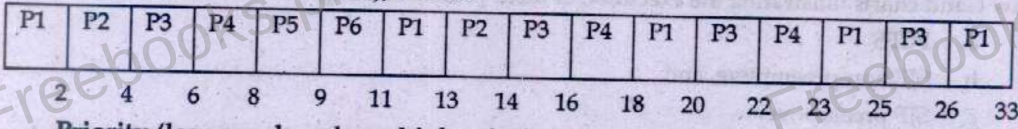
ATT=(33+6+18+11+1+3)/6=12



Round-Robin (quantum=2):

AWT=(18+11+18+8+9)/6=14.66

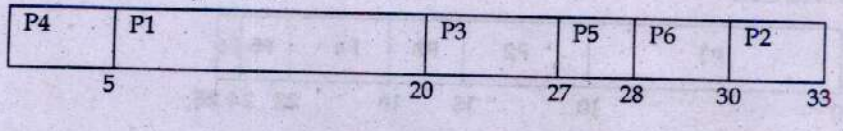
ATT=(33+14+26+23+9+11)/6=19.33



Priority (low numbers have high priority):

AWT=(5+30+20+0+27+28)/6=18.33

ATT=(20+33+27+5+28+30)/6=23.83



Consider the following set of processes:

Arrival Time	Process	Burst Time	Priority
0	P1	8	2
2	P2	6	2
4	P3	3	1
10	P4	3	4

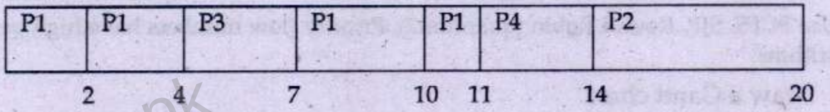
Use the preemptive version of SJF and priority (low numbers have high priority). For each of the algorithms:

- Draw a Gantt chart
- Obtain the average waiting time
- Obtain the average turnaround time

SJF:

AWT=((7-4)+(14-2)+0+(11-10))/4=4

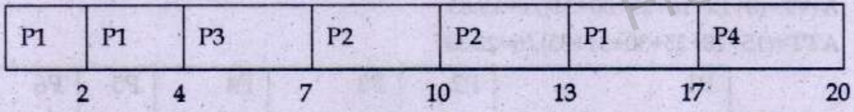
ATT=(11+(20-2)+(7-4)+(14-10))/4=9



Priority (low numbers have higher priority):

AWT=((13-4)+(7-2)+0+(17-10))/4=5.25

ATT= (17+(13-2)+(7-4)+(20-10))/4=10.25



26. Consider the following set of processes with the length of CPU-burst time given in milliseconds. Arrival time is the time at which the process is added to the ready queue.

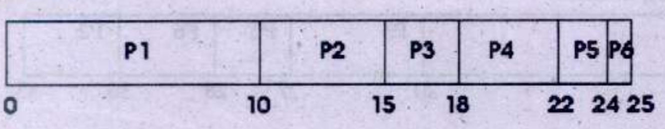
Process	Burst Time	Arrival Time
P1	10	0
P2	5 1	0
P3	3	0
P4	4 3	0
P5	2	4
P6	1 1	6

Draw Gantt charts illustrating the execution of these processes using

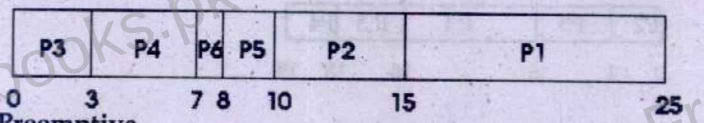
a. FCFS
b. SJF non-preemptive, and
c. SJF preemptive.

Calculate the wait times for each process each strategy. Calculate the average wait times under each strategy.

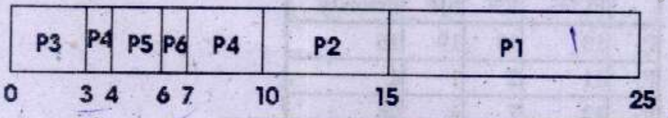
27. Answer: FCFS



SJF non-preemptive


P3
P4
P6
P5
P2
P1
0
3
7
8
10
15
25

SJF Preemptive


P3
P4
P5
P6
P4
P2
P1
0
3
4
6
7
10
15
25

	FCFS	SJF	SJF (preemptive)
P1	0	15	15
P2	10	10	10
P3	15	0	0
P4	18	3	6
P5	18	4	0
P6	18	1	0
Average	13.17	5.5	5.17

28. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

a. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum=1) scheduling.

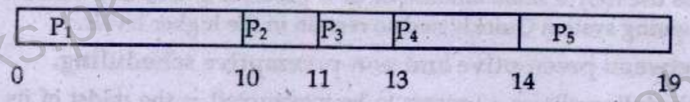
b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

c. What is the waiting time of each process for each of the scheduling algorithms in part a?

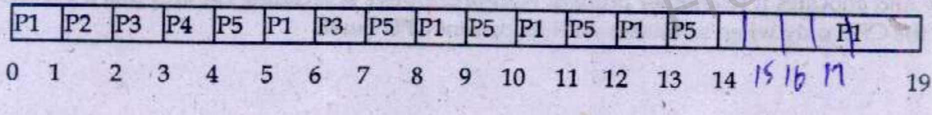
d. Which of the schedules, in part a, results in the minimal average waiting time (over all processes)?

Answer: a. The four Gantt charts are:

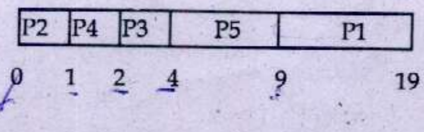
FCFS:


P1
P2
P3
P4
P5
0
10
11
13
14
19

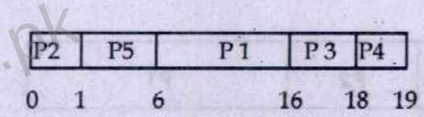
RR:


P1
P2
P3
P4
P5
P1
P3
P5
P1
P5
P1
P5
P1
P5
P1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
19

SJF:


P2
P4
P3
P5
P1
0
1
2
4
9
19

Priority:



b. Turnaround time

	FCFS	RR	SJF	Priority
P_1	10	19	19	16
P_2	11	2	1	1
P_3	13	7	4	18
P_4	14	4	2	19
P_5	19	14	9	6

c. Waiting time (turnaround time minus burst time)

	FCFS	RR	SJF	Priority
P_1	0	9	9	6
P_2	10	1	0	0
P_3	11	5	2	16
P_4	13	3	1	18
P_5	14	9	4	1

d. Shortest Job First

29. What is meant by the term dispatch latency?

This is the time it takes for the dispatcher to stop one process and start another running.

30. What are various criteria must be considered in making a good scheduling algorithm?

Fairness: Make sure each process gets its fair share of the CPU

Efficiency: Keep the CPU busy 100 percent of the time

Response time: Minimize response time for interactive users

Turnaround: Minimize the time batch users must wait for output

Throughput: Maximize the number of jobs processed per hour

31. A preemptive scheduling method has higher overhead than a non-pre-emptive method. Why then do all interactive systems use preemptive methods?

One cannot provide reasonable service to short jobs without preemption. They languish behind long ones. Interactive processes are short processes in the short-term scheduler.

32. How does multilevel feedback (MLFB) scheduling policy favor I/O bound jobs to get good I/O device utilization?

I/O bound jobs use only a small amount of CPU processing time and then block for I/O. Hence they will leave the queuing system Quickly and so remain in the higher levels.

33. Differentiate between preemptive and non-preemptive scheduling.

Preemptive scheduling allows a process to be interrupted in the midst of its execution. It takes CPU away and allocates it to another process. Non-preemptive scheduling ensures that a process leaves control of the CPU only when it finishes with its current CPU burst.

5 PROCESS SYNCHRONIZATION

Chapter Overview

- 5.1 Process Synchronization
- 5.2 Race Conditions
- 5.3 Critical-Section Problem
    - 5.3.1 Criterion to Solve Critical-Sections Problem
    - 5.3.2 Two-Process Solution for Critical Section
        - 5.3.2.1 Algorithm 1
        - 5.3.2.2 Algorithm 2
        - 5.3.2.3 Algorithm 3
    - 5.3.3 Bakery Algorithm Multiple-Process Solution
- 5.4 Hardware Solution to Synchronization (Test & Set)
- 5.5 Semaphores
- 5.6 Classic Problems of Synchronization
    - 5.6.1 Bounded-Buffer Producer-Consumer Problem
    - 5.6.2 Readers-Writers Problem
    - 5.6.3 The Dining Philosophers Problem
    - 5.6.4 Sleeping Barber Problem
- 5.7 Counting Semaphores
- 5.8 Monitors
    - 5.8.1 Advantages of Monitors
    - 5.8.2 Condition Variable
    - 5.8.3 Implementation Questions for Monitors
    - 5.8.4 Readers and Writers Problem using Monitors

Review Questions

5.1 Process Synchronization

Process synchronization deals with various mechanisms to ensure orderly execution of cooperating processes that share a logical address space. The basic purpose of process synchronization is to maintain data consistency. A cooperating process is one that can affect or be affected by other processes executing in the system.

Concurrently running processes without any sort of data protection scheme can manipulate data and produce an incorrect result. Moreover, if the outcome of an execution depends on the particular order in which the data access takes place, a race condition can occur. So we need to implement some sort of synchronization of the processes.

5.2 Race Conditions

The race condition is the situation where several processes access and manipulate shared data concurrently. The final value of the shared data depends upon which process finishes last. To prevent race conditions, concurrent processes must be synchronized.

Consider the following simple procedure:

```
void deposit(int amount)
{
   balance = balance + amount;
}
```

Where we assume that **balance** is a shared variable. If two processes try to call the process **deposit** concurrently, something very bad can happen. The single statement "balance = balance + amount" is actually implemented by a sequence of instructions such as:

```
Load    Reg, balance     // loads the previous balance in register
Add     Reg, amount      // adds the amount
Store   Reg, balance     // updates the calculated amount back in balance
```

Suppose process P1 calls *deposit (10)* and process P2 calls *deposit (20)*. If one completes before the other starts, the combined effect is to add 30 to the balance. However, the calls may happen at exactly the same time. Suppose the initial balance is 100, and the two processes run on different CPUs. One possible result is as follows:

P1 loads 100 into its register
P2 loads 100 into its register
P1 adds 10 to its register, giving 110
P2 adds 20 to its register, giving 120
P1 stores 110 in balance
P2 stores 120 in balance

and the final effect is to add only 20 to the balance.

This kind of bug is called a **race condition**. It only occurs under certain timing conditions. It is very difficult to track down it since it may disappear when you try to debug it. It may be nearly impossible to detect from testing since it may occur very rarely. The only way to deal with race conditions is through very careful coding. The systems that support processes contain constructs called **synchronization primitives** to avoid these problems.

5.3 Critical-Section Problem

A section of code or collection of operations in which only one process may be executing at a given time, is called **critical section**. Consider a system containing **n** processes { $P_0, P_1, P_2, \ldots, P_n$ }. Each process has a segment of code called a **critical section** in which the process may be changing common variables, updating a table, writing into files etc. When such a system works, only one process may be allowed to execute within a critical section. The execution of critical sections by the processes is **mutually exclusive** in time.

The **critical-section** problem is to design a protocol that the processes can cooperate. Each process must request permission to enter its critical section. The section of code implementing this request is called **entry section**. The critical section may be followed by a section of code known as **exit section**. The remaining code is known as **remainder section**.

Example

Here is a general structure of a typical process:

```
while (1)
{
    entry section;
    critical section;
    exit section;
    remainder section;
}
```

Section Of Process

5.3.1 Criterion to Solve Critical-Sections Problem

Any solution to critical section problem must satisfy the following three requirements:

1. Mutual Exclusion

If a process P_i is executing in its critical section, then no other processes can be executing in their critical sections.

2. Progress

If no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing their remainder section can participate in the decision on which will enter its critical section next. This selection cannot be postponed indefinitely.

3. Bounded Waiting

There exists a bound on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

Note: No assumptions can be made about speeds, No. of CPUs and No. of processes.

5.3.2 Two-Process Solution for Critical Section

First of all, we will consider the algorithms that can be applied only on two processes at a time. The processes are P_i and P_j (where i = 0 and j = 1).

5.3.2.1 Algorithm 1

Suppose a shared variable **s** is initialized to 0 or 1. If **s** is 0, then P_0 can execute its critical section and if **s** is 1, then P_1 can execute. This algorithm ensures that there is only one process in its critical section at a time.

Here is a representation of this algorithm for P_i.

```
while (1)
{
    while (s != i);
                critical section;
    s = j;
                remainder section;
}
```

Now, let's see the representation of the same algorithm for P_j.

```
while (1)
{
    while (s != j);
                critical section;
    s = i;
                remainder section;
}
```

How it Works?

Suppose, s is initialized to 1 so P_0 cannot enter its critical section as only P_1 is allowed to enter its critical section at start. P_0 can enter its critical section only if the value of s is 0. s is set to 0 by P_1 when it executes its exit section. In this way, mutual exclusion is guaranteed.

Problem in Algorithm 1

In this algorithm, the progress requirement is not satisfied. For example, if s is set to 1 and P_0 is ready to enter its critical section, it cannot enter even if P_1 is in its remainder section. P_0 has to wait until P_1 enters its critical section, executes it and then sets the value of s to 0.

5.3.2.2 Algorithm 2

Algorithm 1 only remembers which process is allowed to enter its critical section. It does not know any other information about the processes. We can use a boolean array **flag[2]** initialized to **false**. If flag[i] is **true**, it indicates that Pi is ready to enter its critical section.

The structure of P_i in this algorithm is as follows:

```
while (1) {
    flag[i] = true;
    while (flag[j]);
                critical section;
    flag[i] = false;
                remainder section;
}
```

The structure of Pj for this algorithm is as follows:

```
while (1) {
    flag[j] = true;
    while (flag[i]);
                critical section;
    flag[j] = false;
                remainder section;
}
```

How it Works?

First, Pi sets flag[i] to **true** that indicates that Pi is ready to enter its critical section. Secondly, it checks whether Pj wants to enter its critical section. It Pj wants, then Pi will wait until Pj exits its critical section and sets flag[j] to **false**. Now, Pi can enter its critical section. It will set flag[I] to **false** thus allowing Pj to enter its critical section again, if needed.

Problem in Algorithm 2

This algorithm depends on the timing of the processes. Suppose, P_0 sets flag[0] to true and context switch occurs and shifts the control to P_1. Now, P_1 also sets flag[1] to true. At this point, both processes will wait for the other and none will be able to enter its critical section.

5.3.2.3 Algorithm 3

We can use a third algorithm by combining the previous two algorithms. It provides a solution for critical section problem. In this algorithm, both flag[2] and **s** are used. The structure of this algorithm for Pi is as follows:

```
while (1)
{
   flag[i] = true;
   s = j;
   while (flag[j] && s == j);
                  critical section;
   flag[i] = false;
                  remainder section;
}
```

The structure of this algorithm for Pj is as follows:

```
while (1) {
   flag[j] = true;
   s = i;
   while (flag[i] && s == i);
                  critical section;
   flag[j] = false;
                  remainder section;
}
```

How it Works?

Suppose, both flag[0] and flag[1] are set to **false**. If Pi tries to enter its critical section, it sets flag[i] to **true**. Then it sets s to j to ensure that Pj may enter its critical section, if needed. If both processes try to enter critical sections, both will set their corresponding **flag** value to **true**. Then, Pi will set *s* to j and Pj will set s to i. The value of s will then decide which process can enter its critical section. In this solution:

- Mutual exclusion is guaranteed
- Progress requirement is satisfied
- Bounded-waiting is met

Proof

Mutual Exclusion

Pi can enter its critical section only if flag[j] is false or **s** = **i**. If both processes try to enter critical sections, the value of **s** will decide it to ensure mutual exclusion.

Progress & Bounded-waiting Only while loop of **Pi** can stop it from entering its critical section **while (flag[j] && s == true**. If **Pj** is not ready to enter its critical section and **flag[j]** is false, **Pi** can enter its critical section. If **Pj** has set **flag[j]** to true and is executing its while statement, then the value of **s** will decide which process can enter its critical section. Suppose **Pj** enters its critical section. When it exits its critical section, it will set **flag[j]** to false and **Pi** will be able to enter its critical section. If **Pj** sets **flag[j]** to true, it should also set turn **s** to i. As **Pi** does not change the value of **s** in while statement, it will enter its critical section after a maximum of one entry by **Pj** (bounded-waiting).

5.3.3 Bakery Algorithm Multiple-Process Solution

This algorithm solves the critical section problem for **n** processes. The basic idea is that of a bakery; customers take numbers, and the customer with the lowest number gets service next. Here, of course, "service" means entry to the critical section. The structure of Pi for bakery algorithm is as follows:

```
while (1)
{
   choosing[i] = true;
   number[i] = max(number[0],number[1], ... , number[n-1]) + 1;
   choosing[i] = false;
   for (j=0; j<n; j++)
   {
   while (choosing[j];
   while ((number[j] != 0) && (number[j,j] < number[i,i]));
   }
                  critical section;
   number[i] = 0;
                  remainder section;
}
```

How it Works?

First, choosing[i] is **true** if Pi is choosing a number. The number that Pi will use to enter the critical section is in number[i]. It is 0 if Pi is not trying to enter its critical section. First three lines indicate that the process is choosing a number, and then tries to assign a unique number to the process Pi; but it doesn't always happen.

In for loop, we select which process goes into the critical section. Pi waits until it has the lowest number of all the processes waiting to enter the critical section. If two processes have the same number, the one with the smaller name - the value of the subscript - goes in. The notation "(a,b) < (c,d)" means **true** if a < c or if both a = c and b < d. Note that if a process is not trying to enter the critical section, its number is 0. Also, if a process is choosing a number when Pi tries to look at it, Pi waits until it has done so before looking.

After exiting from its critical section, Pi is no longer interested in entering its critical section, so it sets number[i] to 0.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

5.4 Hardware Solution to Synchronization (Test & Set)

The critical problem can be solved in a single-processor system at hardware level. We can stop interrupts to occur during modification of a shared variable so that the instruction can be executed without preemption. This solution is not applicable in multi-processor system because disabling of interrupts is time-consuming.

Many machines provide hardware instruction to test and modify a word and swap the contents of words atomically without any interruption. The TestAndSet instruction is executed atomically.

The definition of **TestAndSet** is as follows:

```
boolean TestAndSet (boolean &target)
{
   boolean rv = target ;
   target = true;
   return rv;
}
```

A machine that supports TestAndSet instruction, can implement mutual exclusion by using a Boolean variable lock, initialized to false. The structure of such implementation is as follows:

```
while (1)
{
   while (TestAndSet(lock));
                  critical section;
   lock = false;
                  remainder section;
}
```

The Swap instruction is also executed atomically and operates on two word to interchange their contents.

The definition of **Swap** is as follows:

```
void Swap(boolean &a, boolean &b)
{
   boolean temp = a;
   a = b;
   b = temp;
}
```

The implementation of **TestAndSet** is as follows:

```
while (1)
{
   waiting[I] = true;
   key = true;
   while (waiting[I] && key)
             key = TestAndSet(lock);
   waiting[I] = false;
                          critical section;
   j = (I+1) % n;
```

```
    while ((j != I) && !waiting[j])
                j = (j+1) % n;
    if (j == I)
                lock = false;
    else
                waiting[j] = false;
}
```

Proof

Mutual Exclusion	Here, **Pi** can enter its critical section only if **waiting[i]** is true or key is false. The value of key can be false only if TestAndSet instruction is executed. When a process executes, it will find key = false. The remaining processes will have to wait. The value of **waiting[i]** can be false only if another process leaves its critical section. In this way, only one **waiting[i]** is false. It maintains mutual exclusion.
Progress	A process that leaves its critical section either sets lock to false or sets **waiting[j]** to false. It allows a waiting process to enter its critical section.
Bounded Waiting	A process that leaves its critical section, visits the array in a cycle and finds out any process with **waiting[j] = true**. It designates that process to enter its critical section. The selected process will also do so when leaving its critical section.

5.5 Semaphores

Semaphores provide a more organised way of controlling the interaction of multiple processes than simple variables. A semaphores is an integer variable used by processes to send signals to other processes. It can only be accessed by the following two operations:

- P (Wait or Down)
- V (Signal or Up)

The definition of **wait** is as follows:

```
Wait (S)
{
    while (S <= 0); // do nothing
    S--;
}
```

The definition of **signal** is as follows:

```
Signal (S)
{
    S++;
}
```

Any change in the value of semaphore, in wait and signal operations, must be executed indivisibly. If one process is changing the value of a semaphore, no other process is allowed to make any change simultaneously.

Use of Semaphores

Semaphores are used to solve the critical section problem for **n** processes. The processes use a shared semaphore **mutex** initialized to 1. The implementation of semaphore for mutual exclusion is as follows:

```
while (1)
{
   wait (mutex);
              critical section;
   signal (mutex);
              remainder section;
}
```

How it Works?

Here, **mutex** is a shared semaphore that is initialized to 1. Suppose P0 executes wait operation on semaphore. It can enter its critical section because the value of mutex is 1. It will be decreased by 1 and P0 will enter its critical section. At this point, if P1 also tries to enter its critical section and executes **wait** operation, it will have to wait because the value of mutex is now 0. It will wait until P0 execute signal operation and increases the value of mutex by 1. So, only one process at a time can enter its critical section, which guarantees mutual exclusion.

5.6 Classic Problems of Synchronization

Following are the classic problems of synchronization in operating system:

5.6.1 Bounded-Buffer Producer-Consumer Problem

Suppose there are **producer** and **consumer** processes. Producer produces objects, which consumer uses for something. There is one **Buffer** object used to pass objects from producers to consumers. A Buffer is used to store production of producer.

The problem is to allow producers and consumers to access the Buffer while ensuring the following:

1. The shared Buffer should not be accessed by these processes simultaneously.
2. Consumers do not try to remove objects from Buffer when it is empty.
3. Producers do not try to add objects to the Buffer when it is full.

When condition 3 is dropped (the **Buffer** can have infinite capacity), the problem is called the **unbounded-buffer problem**, or sometimes just the **producer-consumer problem**.

Producer process

```
while(1)
{
   produce an item in nextp;
   wait (empty);
   wait (mutex);
   add nextp to buffer
   signal (mutex);
   signal (full);
}
```

Consumer process

```
while(1)
{
  wait( full );
  wait( mutex );
  remove an item from buffer to nextc
  signal( mutex );
  signal( empty );
  consume the item in nextc;
}
```

How it Works?

Here, buffer is the data structure for storing the production of producer process. **mutex** is a semaphore for providing mutual exclusion and is initialized to 1. Two semaphores **empty** and **full** are used to represent the number of full and empty buffers. The semaphore empty is initialized to n and full to 0.

- The producer puts data in the buffer.
- The buffer holds only n items.
- The consumer gets data from the buffer.
- The producer works only if the buffer is not full.
- The consumer works only if the buffer is not empty.

5.6.2 Readers-Writers Problem

In this problem, a number of concurrent processes require access to some object (such as a file.) Some processes extract information from the object and are called **readers**. Some processes change or insert information in the object and are called **writers**. The **Bernstein condition** states that many readers may access the object concurrently, but if a writer is accessing the object, no other processes (readers or writers) may access the object.

There are two possible policies for doing this:

First Readers-Writers Problem Readers have priority over writers. It means that unless a writer has permission to access the object, any reader requesting access to object will get it. This may result in a writer waiting indefinitely to access the object.

Second Readers-Writers Problem Writers have priority over readers. It means that when a writer wishes to access the object, only readers, which have already obtained permission to access the object, are allowed to complete their access. Any readers that request access after the writer has done so must wait until the writer is done. This may result in readers waiting indefinitely to access the object.

There are two concerns:

- Enforce the Bernstein conditions among the processes
- Enforce the appropriate policy of whether the readers or the writers have priority.

A typical example of this occurs with databases, when several processes are accessing data. Some will want only to read the data, others to change it. The database must implement some mechanism that solves the readers-writers problem.

Shared Data

```
semaphore mutex = 1;
semaphore wrt = 1;
int readcount = 0;
```

Writer Process

```
wait(wrt);
writing is performed
signal (wrt);
```

Reader Process

```
wait (mutex);
readcount = readcount + 1;
if (readcount ==1)
  wait (wrt);
signal (mutex);
reading is performed
wait(mutex);
readcount = readcount - 1;
if (readcount == 0)
  signal (wrt);
signal (mutex);
```

5.6.3 The Dining Philosophers Problem

There are five philosophers seated around a dining table, with a plate of food in front of each one. On the table between each pair of philosophers there is a single chopstick.

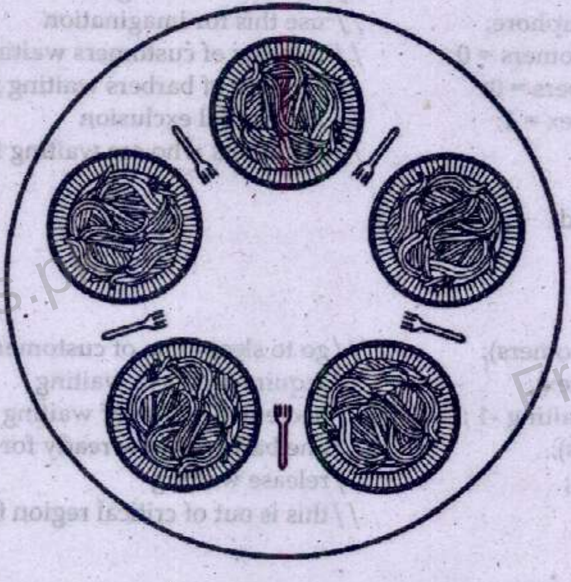


Figure: The Dining Philosopher's Table

A philosopher needs to pick up both the left and right chopstick in order to eat. If either is in use by another philosopher, he must wait. The problem is to find a way to synchronize access to the chopsticks to make sure that every philosopher gets a chance to eat.

One solution is to control access to each chopstick with a separate semaphore. When a philosopher wants to pick up a chopstick, it performs **P** operation on the corresponding semaphore. When chopstick is put back on table, **V** is performed on the semaphore. Suppose the philosophers are numbered 0-4 and semaphores are also numbered from 0 to 4. For philosopher **i**, the left chopstick is chopstick **i** and the right chopstick is chopstick [(i+1)%5].

Shared Data

```
semaphore chopstick[5];
```

Philosopher i:

```
while(1)
{
 wait (chopstick[i]);
 wait (chopstick[i+1 % 5]);
 eat;
 signal (chopstick [1]);
 signal (chopstick [i+1 % 5]);
 think;
}
```

5.6.4 Sleeping Barber Problem

A barber shop consists of a waiting room with chairs and the barber room contains the barber chair. The barber goes to sleep if there are no customers. If a customer enters the barber shop and all chairs are occupied, the customer leaves the shop. The customer sits in a chair if the barber is busy but chairs are available. The customer wakes up the barber if he is asleep. The following is a sample solution for the sleeping barber problem.

```
# define CHAIRS 5                  // chairs for waiting customers
typedef int semaphore;             // use this for imagination
semaphore customers = 0;           // number of customers waiting for service
semaphore barbers = 0;             // number of barbers waiting for customers
semaphore mutex = 1;               // for mutual exclusion
int waiting = 0;                   //customers who are waiting for a haircut

void barber(void)
{
  while (TRUE)
  {
    down(&customers);              //go to sleep if no of customers are zero
    down(&mutex);                  //acquire access to waiting
    waiting = waiting -1 ;         //decrement count of waiting customers
    up(&barbers);                  //one barber is now ready for cut hair
    up(&mutex);                    //release waiting
    cut_hair();                    //this is out of critical region for hair cut
  }
}
void customer(void)
```

```
    {
    down(&mutex);                //enter critical region
    if (waiting < CHAIRS)        //if there are no free chairs, leave
    {
        waiting = waiting +1;    //increment count of waiting customers
        up(&customers);          //wake up barber if necessary
        up(&mutex);              //release access to waiting
        down(&barbers);          //go to sleep if no of free barbers is zero
        get_haircut();           //be seated and be serviced
    }
    else
    {
    up (&mutex);                 // shop is full: do no wait
    }
  }
```

Explanation

This problem is similar to various queuing situations. It should manage the barber and the customers without getting into race conditions. The solution uses three semaphores:

- **customers is used to counts the waiting customers**
- **barbers** is the number of barbers (0 or 1)
- **mutex** is used for mutual exclusion

It also needs a variable **waiting** to count the waiting customers. The barber executes the procedure **barber** to block the semaphore **customers** with the initially value of 0. The barber then goes to sleep. The customer procedure is executed when a customer arrives starting by acquiring **mutex** to enter a critical region. If another customer enters in the meantime, the second one will not be able to do anything until the first one has released **mutex**. The customer checks to see if the number of waiting customers is less than the number of chairs. He releases **mutex** and leaves without a haircut if no chair is free. The customer increments the integer variable **waiting** if a chair is available. He also performs **up** operation on the semaphore **customers**. The barber begins the haircut when the customer releases mutex.

5.7 Counting Semaphores

Counting semaphores are used when a resource is to be allocated from a pool of identical resources. The semaphores initialized the number of resources in the pool. When a resource is allocated and **p** operation is executed, it decrement semaphores by 1. When **v** operation is executed, it increments semaphores by 1. It indicates that the resource has been released and is available for another process. If p operation is called when the semaphore has been decremented to zero, the process has to wait until a resource is released by V operation.

5.8 Monitors

A **monitor** is a concurrency construct containing data and procedures required to perform allocation of a particular serially reusable-shared resource or group of resources. For allocation of a resource, a process must call **monitor entry**. More than one process may want to enter the monitor but mutual exclusion is implemented at monitor boundary. Only process at a time can enter a monitor. The remaining processes wait until the monitor becomes empty. This is automatically managed by the monitor and mutual exclusion is guaranteed.

The monitor data may be global to all procedures of that monitor or local to a particular procedure. Data is a accessible within a monitor. It is called **information hiding**.

If a process calling a monitor entry finds the required resource already allocated, the monitor procedure calls *wait* and that process waits outside the monitor for the resource to be released. When the required resource is returned to the system, the monitor entry calls *signal* to allow the waiting process to gain the resource and leave the monitor.

5.8.1 Advantages of Monitors

Some advantages of monitors are as follows:

- A process calling a monitor procedure or method can ignore actual implementation.
- Once a monitor is correctly programmed, it remains correct, despite the number of processes executing *(as in object-oriented programming)*
- The implementation of a monitor can be changed without affecting the application or the user's view of the monitor resources *(as in object-oriented programming)*
- Monitors provide mutual exclusion

5.8.2 Condition Variable

Condition variables allow a process executing within the monitor to be put to sleep to wait for some condition to be set (signaled). They are used to delay a process that cannot safely proceed until there is a change in the state of the monitor. This avoids deadlock within the monitor.

Condition variables can also awaken a sleeping process to let it be actively executing again within the monitor. The condition variables wake up the delayed or suspended processes within the monitor.

A condition variable is just a data structure *(or class)* consisting of the following:

- A Boolean value
- A queue of delayed processes

It is a shared data variable in monitor. Commands related to condition variables are:

wait(c)	The process currently active in the monitor suspends execution and gives up mutual exclusion to the monitor until the condition variable **c** is signaled. It is placed on the end of queue of delayed processes waiting for **c** to be signaled.
signal(c)	The process at the front of the queue is awakened and resumes execution within the monitor. If the queue connected to the condition variable **c** is empty, nothing happens. It is equivalent to a skip operation.

5.8.3 Implementation Questions for Monitors

Suppose process **Q** is waiting on the condition variable **c** in a monitor. Further suppose that process **P** is active in the monitor and executes *c.signal*, waking up **Q**.

Now which process continues to be active in the monitor? (Remember that only one process can be active at any time.)

This turns out to be an implementation issue (i.e. how the monitors are implemented). When **P** signals **Q**, there are three choices of action:

- **P** may continue to execute in the monitor. However, if it does so, **P** may alter the condition that awakened **Q**.
- **P** may wait (suspend) while **Q** executes in the monitor until **Q** is done or some other condition becomes true.

- **P** executes the *signal* command and immediately leaves the monitor. In other words, the *signal* command is the last line of the procedure **P** executes.

5.8.4 Readers and Writers Problem using Monitors

In this example, **read** and **write** functions used by processes that access database are in a monitor *ReadersWriters*. If a process wants to write to database, it must call *writeDatabase* function. If a process wants to read from the database, it must call *readDatabase* function.

Monitors use the primitives **Wait** and **Signal** to put processes to sleep and to wake them up again. In *writeDatabase*, the calling process will be put to sleep if the number of reading processes, stored in the variable *count*, is not zero. Upon exiting the *readDatabase* function, reading processes check to see if they should wake up a sleeping writing process.

Example

```
monitor ReadersWriters
 condition OKtoWrite, OKtoRead;
 int ReaderCount = 0;
 Boolean busy = false;

procedure StartRead()
{
 if (busy)                                  // if database is not free, block
  OKtoRead.wait;
 ReaderCount++;                             // increment reader ReaderCount
 OKtoRead.signal();

}
procedure EndRead()
{
 ReaderCount-- ;                            // decrement reader ReaderCount
 if ( ReaderCount == 0 )
  OKtoWrite.signal();
}
procedure StartWrite()
{
 if ( busy || ReaderCount != 0 )
  OKtoWrite.wait();
 busy = true;
}

procedure EndWrite()
{
 busy = false;
 If (OKtoRead.Queue)
  OKtoRead.signal();
 else
  OKtoWrite.signal();
}
Reader()
```

```
{
  while (TRUE)                        // loop forever
  {
   ReadersWriters.StartRead();
      readDatabase();                 // call readDatabase function in monitor
   ReadersWriters.EndRead();
  }
}

Writer()
{
  while (TRUE)                        // loop forever
  {
       make_data(&info);              // create data to write
   ReaderWriters.StartWrite();
       writeDatabase();               // call writeDatabase function in monitor
   ReadersWriters.EndWrite();
  }
}
```

Review Questions

1. What is a critical section?

A section of code that involves shared resources that we must protect with mutual exclusion so that only one thread at a time can execute that section of code.

2. What is a race condition? How can we prevent a race condition?

The race condition is the situation where several processes access and manipulate shared data concurrently. The final value of the shared data depends upon which process finishes last. To prevent race conditions, concurrent processes must be synchronized.

3. What are three requirements of good solution to the critical-section problem?

1) Mutual exclusion is guaranteed

2) Progress is maintained. Processes not in their critical sections may not block other processes from entering their critical sections.

3) Bounded waiting is assured

4. Consider the following solution to the critical-section (mutual exclusion) problem. Consider the case of 2 processes, P0 and P1.

```
shared int turn = 0;
P0:                          P1:
while (turn !=0) ;       while (turn != 1) ;
critical section             critical section
turn = 1;                    turn = 0;
```

5. Is this a good solution to the critical-section problem? What problems are involved?

It is not good solution because this solution may violate the progress requirement. Since the processes must strictly alternate entering their critical sections, a process wanting to enter its critical section twice in a row will be blocked until the other process enters and leaves its critical section.

6. What is a semaphore?

A semaphore S is an integer variable that can only be accessed via two atomic operations:

```
wait(S): while (S <= 0) /* do nothing */ ;
         S = S - 1;
signal(S): S = S + 1;
```

7. Consider the following semaphore definition:

A semaphore S is an integer variable that can only be accessed via two atomic operations

```
wait(S): while (S <= 0) /* do nothing */ ;
     S = S - 1;
signal(S): S = S + 1;
```

A solution to the mutual exclusion problem, using these operations is shown below:

```
shared int S = 1; /* S is initialized to 1 */
```

Each process executes the following code:

```
while (TRUE) {
  wait(S);
   critical section
  signal(S);
}
```

8. a. Does the above solution involve any busy-waiting? Explain.

b. What is the "priority inversion" problem?

c. If priority scheduling is used, can the above solution lead to the priority inversion problem? If so, give an example; otherwise, explain why not.

a. This semaphore definition involves the busy waiting because a waiting process is looping for the shared variable S.

b. A priority inversion problem is a situation in which a low-priority process is blocking a high-priority process.

c. Yes, the above solution can lead to the priority inversion problem if a high-priority process is waiting for a low-priority process to leave the critical section and signal the shared variable S.

9. Suppose that a process executes the following code:

```
semaphore S_a = 0;
semaphore S_b = 0;
wait(S_a);
 - access resource R_a
signal(S_a);
wait(S_b);
 - access resource R_b
signal(S_b);
```

10. Briefly explain what happens. Assume that the system has one resource of type R_a and one resource of type R_b. How would you fix the code so that it provides mutually exclusive access to R_a and R_b.

This process will wait on the semaphore S_a and make no progress because the initial value of the semaphore S_a is 0. We can fix it by setting the initial value to 1.

```
semaphore S_a = 1;
semaphore S_b = 1;
```

11. Add semaphores necessary to synchronize processes A, B, C, and D so that A completes before any other process starts, and B completes before C or D may execute, but C and D may execute concurrently. Show your solution.

```
semaphore S1, S2, S3 = 0, 0, 0;
Process A:
----------
- do work of A
signal(S1);     /* Let B start */
Process B:
----------
wait(S1);      /* Block until A is finished */
- do work of B
signal(S2);     /* Let C start */
signal(S3);     /* Let D start */
Process C:
----------
wait(S2);      /* Block until B is finished */
- do work of C
Process D:
----------
wait(S3);      /* Block until B is finished */
- do work of D
```

12. When can the deadlock among processes occur? When can the starvation among processes occur?

1) Two or more processes are waiting indefinitely for an event that can be caused by only one of the waiting processes.

2) A process may never be removed from the queue in which it is suspended.

13. Consider the following proposed solution to the dining philosophers problem:

```
semaphore fork[5] = 1,1,1,1,1; /* each semaphore is initialized to 1 */
philosopher (int i) /* philosopher i=0,1,2,3,4 executes this code */
{
 while(1) {
  think();
  wait(chopstick[i]);
  wait(chopstick[(i+1)%5]);
   eat();
  signal(chopstick[(i+1)%5]);
  signal(chopstick[i]);
 }
}
```

14. Briefly explain how the above solution may lead to deadlock. Give one possible remedy to the deadlock problem.

1) Suppose all philosophers execute the first wait/DOWN operation, before any have a chance to execute the second wait/DOWN operation; that is, they all grab one chopstick. Then, deadlock will occur and no philosophers will be able to proceed. This is called a circular wait.

2) Other Solutions:

a) Only allow up to four philosophers to try grabbing their chopsticks.

b) Pick-up the forks only if both are available. Note: this solution may lead to starvation.

c) Asymmetric solution: Odd numbered philosophers grab their left chopstick first, whereas even numbered philosophers grab their right chopstick first.

15. What is a critical region construct? How does it work?

a. A critical region construct requires that a shared variable v of type T, is declared as:

v: shared T

The variable v can be accessed only inside statement:

region v when B do S where B is a boolean expression.

b. The statement S is executed only when B is true and no other process is the region associated with v. In the way, we can ensure a mutual exclusion solution.

16. What is a monitor? How does it work?

a. A monitor is a collection of procedures, variables, and data structures that can only be accessed by one process at a time.

b. Condition variables are used to allow processes to block themselves and leave the monitor when they cannot proceed. Assume a condition variable x. There are two operations on a condition variable x:

x.wait() - block on condition variable x, and

x.signal() - wakeup a process blocked on x,

Processes are associated with these condition variables to synchronize their operations.

17. Name one similarity and one difference between a semaphore and a monitor.

They both implement mutual exclusion and synchronization. Semaphores do both with a single mechanism where monitors have two mechanisms - one for mutual exclusion (lock) and one for synchronization (condition variables).

18. A north-south highway with two lanes of traffic, one traveling in each direction crosses a river at a bridge. There is only a single lane on the bridge. The bridge is long enough for several cars on the bridge at the one time provided that they are all traveling in the same direction. Write code to be performed when a car wishes to enter or exit bridge, for cars traveling in each direction, using (a) semaphores, (b) monitors.

The bridge is the non-sharable resource. This problem is like a readers and writers problem except that it is more symmetrical (e.g. southbound traffic is not limited to only one car on the bridge at a time). The car nearest the bridge will wait until that car can obtain exclusive access to the bridge for traveling in its direction. Once one northbound car is on the bridge, other cars also traveling north can enter the bridge because the first car will obtain mutual exclusion for north bound cars. When the last northbound car has crossed the bridge, the exclusive use of the bridge will be released. The same applies to southbound traffic.

(a)

```
int  northcars = 0;
int  southcars = 0;
sem northmutex = 1;
sem southmutex = 1;
sem bridge = 1;
/* northbound car arrive at bridge */
wait(northmutex);
if (northcars == 0)
      wait(bridge);
northcars++;
signal(northmutex);
/* enter the bridge */
/* leave the bridge */
wait(northmutex);
northcars--;
```

```
if (northcars == 0)
     signal(bridge);
signal(northmutex);
/* southbound car: symmetrical to northbound car */
```

(b)

```
cond          go_north;
cond          go_south;
int  north_cars = 0;
int  south_cars = 0;
/* northbound car */
northbound_enter()
{
     if (southcars > 0)
               wait(go_north);
     northcars++;
     signal(go_north);
}
northbound_exit()
{
     northcars--;
     if (northcars == 0)
               signal(go_south);
}
/* southbound car: symmetrical to northbound car */
```

19. What are the relative advantages and disadvantages of using semaphores and monitors for implementing process synchronization?

Semaphores:

- They are easy and cheap to implement.
- It is easy to make mistakes, which lead to time dependent errors.
- It is difficult to detect time dependent errors because the behavior of the system is non-deterministic and therefore not easily reproducible.
- A single misbehaving process can crash the system.

Monitors

- Coupled to the use of abstract data types.
- Implemented by compiler: less work for the programmer.
- Implemented by compiler: greater reliability.
- The use of condition variables is sometimes not intuitive.
- Few languages provide monitors: Concurrent Pascal, Concurrent Euclid, Mesa. Ada provides a related construct.

20. The scenario in previous question assumes that all of the philosophers pick up their right fork first. If one of the philosophers were to attempt to pick up their left fork first, and then their right fork, would it still be possible for deadlock to occur?

If one philosopher is left-handed, then deadlock cannot occur, because the "resources" (forks) are always allocated in a particular order, denying the "circular wait" condition for deadlock.

Suppose philosopher 3 is the left-handed one. The philosophers attempt to acquire the resources in the following manner.

Philosopher 1: fork 1, then fork 5

Philosopher 2: fork 2, then fork 1

Philosopher 3: fork 2, then fork 3

Philosopher 4: fork 4, then fork 5

Philosopher 5: fork 5, then fork 4

The "resources" can be regarded as having the order:

fork 2, fork 1, fork 5, fork 4, fork 3

Since no attempt is made to acquire resources except as permitted by this ordering, deadlock cannot occur.

21. It is not possible to implement the producer/consumer problem with one producer and two consumers without using synchronization techniques. Why?

The two consumers will has to share a pointer to the item to consume next and concurrently update this pointer thus introducing a race condition.

22. Compare hardware and software synchronization techniques.

Hardware synchronization primitives are atomic operations implemented within the instruction set of many processors to facilitate the implementation of synchronization. These primitives such as Test-and-Set or Swap allow a lock to be set and tested atomically. Software solutions rely on using a combination of variables such that a particular setting of variables is necessary to allow a process into its critical section. Therefore software solutions tend to involve a large number of variables and are harder to analyze, design and code than hardware solutions.

23. Compare producer – consumer problem and the readers–writers problem.

Both producer-consumer problem and the readers-writers problem are standard synchronization problems. The producer-consumer problem represents concurrent access to data that can be accessed once and should not be shared. The readers-writers problem represents concurrent access to data that can be accessed either in a shared mode (read access) or in an exclusive mode (write access).

24. When are counting semaphores used?

They are implemented when multiple resources are available. When processes request this resource and it is allocated, count is decremented. Once available resources equal 0, requesting processes will be blocked. When resources are de-allocated, count is incremented and can be allocated to any blocked or requesting processes.

25. What is mutual exclusion?

The guarantee that in a shared resource, only one process at a time will have exclusive access to the resource's critical section.

26. What is an atomic operation?

An operation that, once started, completes in a logically indivisible way.

27. What is the test-and-set instruction and what is it used for?

The test-and-set instruction takes a single memory location as an argument and causes the CPU to write this location to a register giving it a value of TRUE without interruption by any other process. The test-and-set instruction allows you to implement simultaneous access to semaphores without the use of interrupts. It can be used along with a yield statement to reduce busy-waiting by allowing the CPU to execute another process while the current process is blocked.

28. What is the purpose for binary and counting semaphore?

1. Binary Semaphore - Only deals with 1 resource - has a value of 0 or 1.

2. Counting Semaphore - deals with more than one resource and uses a lock - when lock is done, it decrements the number of resources by 1, the next process that calls the resource calls lock and the number of resources is decremented again.

29. How are wait/signal operations for monitor different from those for semaphores?

The signal is lost if a process in a monitor signal and no task is waiting on the condition variable. It allows easier program design. In semaphores, every operation affects the value of the semaphore. The wait and signal operations should be perfectly balanced in the program.

6 DEADLOCKS

Chapter Overview

6.1 Deadlock
6.2 Deadlock Examples
6.3 Resources
    6.3.1 Types of Resources
6.4 Deadlock Characterization
6.5 Resource-Allocation Graph
6.6 Methods for Handling Deadlocks
6.7 Deadlock Prevention
    6.7.1 Mutual Exclusion
    6.7.2 Hold and Wait
    6.7.3 No Preemption
    6.7.4 Circular Wait
6.8 Deadlock Avoidance
    6.8.1 Safe State
        6.8.1.1 Safe Sequence
        6.8.1.2 Unsafe Sequence
    6.8.2 Resource-Allocation Graph Algorithm
    6.8.3 Banker's Algorithm
6.9 Deadlock Detection
    6.9.1 Single Instance of Each Resource Type
    6.9.2 Detection-Algorithm Usage
6.10 Deadlock Recovery
    6.10.1 Process Termination
    6.10.2 Resource Preemption

Review Questions

6.1 Deadlock

In a multiprogramming environment, several processes compete for resources. A situation may arise where a process is waiting for a resource that is held by other waiting processes. This situation is called a **deadlock**.

A system has a finite set of resources such as memory, I/O devices, etc. It also has a finite set of processes that need to use these resources. A process that wishes to use any of these resources, makes a request to use that resource. If the resource is free, the process gets it. If it is being used by another process, it waits for it to become free. The assumption is that the resource will eventually become free and the waiting process will then use the resource. But in some situations, the other process may also be waiting for some resource.

"A set of processes is in a deadlock state when every process in the set is waiting for an event that can only be caused by another process in the set."

Suppose we have two tape drives and two processes. Each process has got one tape drive for each process, but both need two such tape drives to proceed with execution. Each is waiting for the other process to release the other tape drive. This will never happen as the other is also waiting for the same thing. This situation is a **deadlock**.

6.2 Deadlock Examples

Following are some examples of deadlock.

Example 1

The following figure is of a traffic deadlock. It makes the nature of deadlock very clear. None of the cars can pass unless at least one of them backs up or is removed.

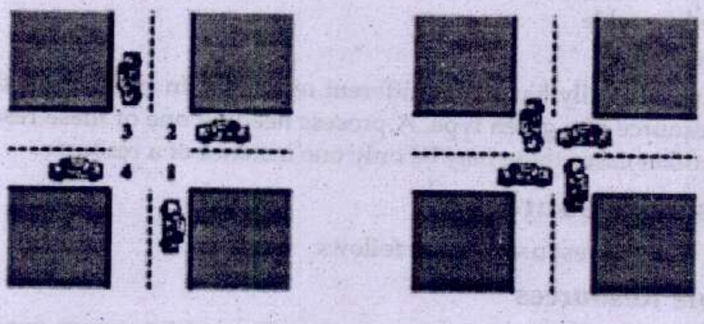


Figure: (a) Deadlock Possible (b) Deadlock Occurred

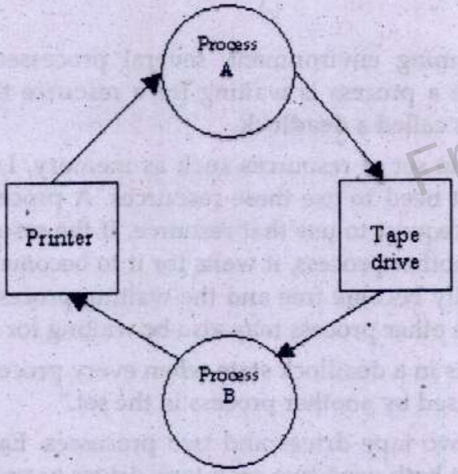
Example 2

When two trains approach each other at a crossing, both shall come to a full stop and neither will start up again until the other has gone.

Example 3

Suppose we have a printer and a tape drive. Process A requests for the printer and gets it. Process B requests tape drive and it is granted to it. Now A asks for the tape drive but the request is denied until B releases it. At this point, the process B asks for the printer before releasing the tape drive.

Now both processes are waiting for each other to release the resource and are blocked. Both processes will remain in this situation forever. This situation is called **deadlock**.



6.3 Resources

A resource is an object that is used by a process. It can be a piece of hardware such as:

- Tape drive
- Disk drive
- Printer

A resource can be a piece of information such as:

- File
- A record within a file
- A shared variable
- A critical section

A computer typically has many different resources. In some cases, there may be many instances of a resource of a given type. A process needing one of these resources can use any one of them. In other cases there may be only one instance of a resource.

6.3.1 Types of Resources

Different types of resources are as follows:

1. Preemptible Resources

A **preemptible resource** is one that can be allocated to a given process for a period of time. Then it can be allocated to another process. Then it can be reallocated to the first process without any negative effects. Examples of preemptible resources include memory, buffers, CPU and array processor.

2. Nonpreemptible Resources

A **nonpreemptible resource** cannot be taken from one process and given to another without side effects. One example is a printer. A printer cannot be take away from one process and given to another process in the middle of a print job. Deadlocks usually involve nonpreemptible resources.

The usual sequence of events that occur as a resource is used is as follows:

1. Request the resource	One of two things can happen when a resource is requested. The request can be granted immediately if it is available. The request can be postponed or blocked until a later time.

2. Use the resource	Once the resource has been acquired, it can be used.
3. Release the resource	A process releases the resource when the process no longer needs it. Usually it is released as soon as possible.

6.4 Deadlock Characterization

Deadlock occurs if the following four conditions take place simultaneously in a system:

1. Mutual Exclusion

At least one resource must be held in a non-sharable mode. It means that only one process at a time can use the resource. If another process requests the resource, the requesting process must be delayed until the resource has been released.

2. Hold and Wait

A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.

3. No Preemption

Resources cannot be preempted. A resource can be released only voluntarily by the process holding it after that process has completed its task.

4. Circular Wait

A set {P_0, P_1, P_2, ..., P_n} of waiting processes must exist such that P_0 is waiting for a resource that is held by P_1, P_1 is waiting for a resource that is held by P_2, ..., P_{n-1} is waiting for a resource that is held by P_n, and P_n is waiting for a resource that is held by P_0.

All four conditions must hold for a deadlock to occur.

6.5 Resource-Allocation Graph

The resource allocation graph is used to describe a deadlock graphically. The graph has two different types of nodes i.e. process nodes and resource nodes. Processes are represented by circles and resources are represented by rectangles. For each instances of a resource, there is a dot in the resource node rectangle. For example, if there are two identical printers, the printer resource will have two dots. The edges among nodes represent resource allocation and request. If edge goes from resource to process node, it indicates that the process has acquired the resource. If edge goes from process node to resource node, it indicates that the process has requested the resource.

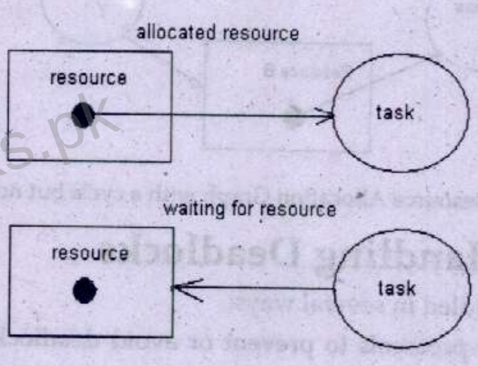


Figure: Resource Allocation Graph

We can use these graphs to determine if a deadlock has occurred or may occur. For example, if all resources only have one instance (all resource node rectangles have one dot) and the graph is circular, a deadlock has occurred. If some resources have several instances, a deadlock may occur. If the graph is not circular, a deadlock cannot occur because the circular wait condition will not be satisfied.

Below is an example of a deadlocked system. Process X has resource B and is waiting for resource A. Process Y has resource A and is waiting for resource B. Note there exists a cycle in the directed graph.

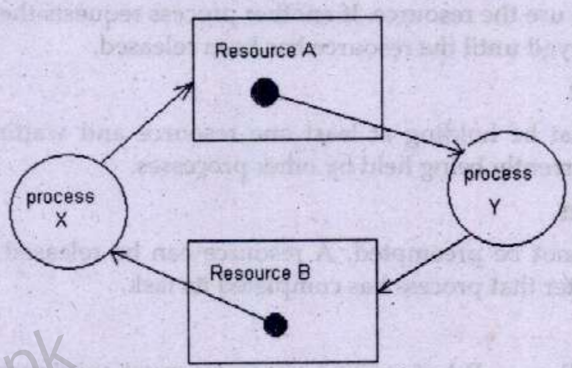


Figure: Resource Allocation showing a deadlock

The diagram below is an example of a resource graph that has a cycle, but does not represent a deadlocked system. It is possible for process Z to continue executing. When process Z is completed, it will release resource A. It will allow process X to acquire it and continue processing. A cycle is a necessary, but not sufficient condition for deadlock. All deadlocked systems will exhibit all of the above four conditions. Not every system that exhibits the above four conditions will be deadlocked.

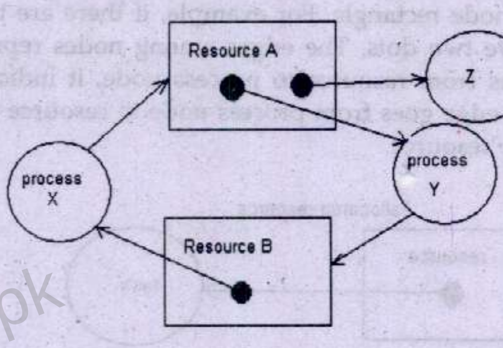


Figure: Resource Allocation Graph with a cycle but not deadlock

6.6 Methods for Handling Deadlocks

A deadlock can be handled in several ways:

1. We can use specific protocols to prevent or avoid deadlocks so that a system may never enter a deadlock state.
2. We can detect the deadlock and recover it.
3. We can totally ignore the deadlock problem.

6.7 Deadlock Prevention

We can prevent deadlocks by ensuring that at least one of the four necessary conditions for deadlock cannot occur. If at least one condition is not satisfied, a deadlock will not occur.

6.7.1 Mutual Exclusion

Mutual exclusion condition must hold for non-sharable resources. For example, only one process can have access to a printer at a time, otherwise the output is disturbed. Some resources can be made sharable like a read-only file. Several processes can be granted read-only access to a file without interfering with each other. However, deadlock cannot be prevented by only denying the mutual exclusion condition because some resources are intrinsically non-sharable.

6.7.2 Hold and Wait

Deadlock can be prevented by denying the hold and wait precondition. This can be implemented in two different ways.

1. One approach is that a process requests all the resources that it needs in one single request at process startup. The system will not grant any resource in the list until it can grant all the required resources.
2. A less restrictive approach is to allow a process to request resources only when it is currently holding no resources. If a process needs a new resource, it must first release all the resources it has and then put the request. It may include a request for the reallocation of a resource it just released.

Problems with This Approach

Each of these strategies has some performance or resource utilization issues.

- If we allocate all resources at the beginning of the process, it may hold resources when it does not need them. It reduces resource utilization. This is especially serious if a process does not know what resources it will actually need for a given execution until it has started working on the data. The process must request all the resources it *might* need.
- A process that needs several popular resources might face **starvation**. Some processes may never execute since some other process always has control of some required resource.

6.7.3 No Preemption

Preemption of resources means that we take away resources from processes when they are waiting for other resources. This could work in the following ways:

1. As soon as a process requests for a resource that is not available, all of its held resources are released. The process is now waiting for all of previously held resources plus the resource that it requested. Suppose a process holds a tape drive and requests a line printer. If line printer is not available, the tape drive is taken away and the process is put into a state of waiting for both a tape drive and a line printer.
2. When a process requests a resource that is available, it gets it. Otherwise, the system checks those processes that are holding the requested resource and may be waiting for some more resources. In this case, the resource is taken away from the waiting process and allocated to the requesting process. If this cannot happen, the process has to wait. During the wait, some other process may get some of its resources.

Problems with This Approach

- It only works if the resources are **preemptible**. Suppose a process has printed output on a line printer and is waiting for some other resource before it can generate more output. The line printer really cannot be taken away without disturbing the output.
- This scheme can also lead to **starvation** for a process that needs several popular resources at the same time. It may keep losing the resources it gets because they do not all become available at the same time.

6.7.4 Circular Wait

We can prevent deadlock by making circular wait impossible. We can define an order by which processes get resources to prevent circular wait. For example, each resource type is assigned a number. The processes can only get resources in increasing order of those resource numbers.

Suppose tape drive has number 1, disk drive has number 5 and printer has number 12. A process wants to read the disk drive and print out the results. It will first need to allocate the disk drive then the printer. It will be prevented from doing it in reverse order.

Problems with This Approach

The problems with this approach are as follows:

- The order of resource numbering may prove arbitrary and inconvenient. This is not very serious problem. There are often natural ways of numbering resources. For example, processes generally use input devices before output devices.
- The order of numbering can force processes to request resources before they need them. It reduces resource utilization. Suppose a process did some work on tape before reading input from a card reader. It would still have to request the reader at the start of processing.

6.8 Deadlock Avoidance

Deadlock avoidance is a technique used to avoid deadlock. It requires the information about how different processes would request different resources. If given several processes and resources, we can allocate the resources in some order to avoid the deadlock.

6.8.1 Safe State

A state is said to be a **safe state** if the system may allocate the required resources to each process up to the maximum required in a particular sequence, without facing deadlock. In safe state, deadlock cannot occur. If a deadlock is possible, the system is said to be in an **unsafe state**. The idea of avoiding a deadlock is to simply not allow the system to enter an unsafe state that may cause a deadlock.

Example

Consider a system with 24 tape drives and three processes: P_0, P_1 and P_2. P_0 may require 20 tape-drives during execution, P_1 may require 8, and P_2 may require up to 18.

Suppose, P_0 is holding 10 tape drives, P_1 holds 4 and P_2 holds 4 tape drives. The system is said to be in a safe state, since there is a safe sequence that avoids the deadlock.

Process	Maximum Required	Current Allocated
P_0	20	10
P_1	8	4
P_2	18	4

6.8.1.1 Safe Sequence

This sequence implies that P_1 can instantly get all of its needed resources. There are 24 total tape drives. P_1 already has 4 and needs 4 more. It can get since there are 6 free tape drives. Once it finishes executing, it releases all 8 resources. At this point, the system has 10 free tape drives. Now P_0 can execute since it requires exactly 10 more drives to finish. After it finishes, the system has a total of 20 free tape drives. These can be allocated to P_2 and it can proceed since it needs 14 more drives to complete its task.

6.8.1.2 Unsafe Sequence

Suppose at some time P_2 requests two more resources to make its holding resources 6. Now the system is in an unsafe state. The system has now 4 free drives and can only be allocated to P_1. After P_1 returns, it will leave 8 free resources. It is not enough for either P_0 or P_2. The system enters a deadlock.

There are algorithms that determine if a safe state can be maintained after a resource allocation. The resource will not be allocated that may cause the system to enter an unsafe state resulting in a deadlock.

6.8.2 Resource-Allocation Graph Algorithm

Resource-allocation graph algorithm is used to avoid deadlock. This algorithm uses the resource-allocation graph by introducing a new edge called **claim edge**. The claim edge is used to indicate that a process may request a resource in future. It is represented by a dashed line. When the process requests that resource, the dashed line is converted to an **assignment** edge i.e. a solid line. It indicates that the resource has been allocated to the process. After the process releases this resource, the assignment edge is again converted to claim edge. All claim edges of a process are represented in the graph. It requires the knowledge of all resources to be used by a process, before the execution of the process starts.

When a process requests a resource, it is allocated only if converting the request edge to an assignment edge does not form a cycle in the graph. If no cycle is formed, the system is in safe state and the resource will be allocated. But if there is a cycle, the system will enter an unsafe state and the process will wait for the resource.

Example

Consider the following figure for the explanation of this algorithm.

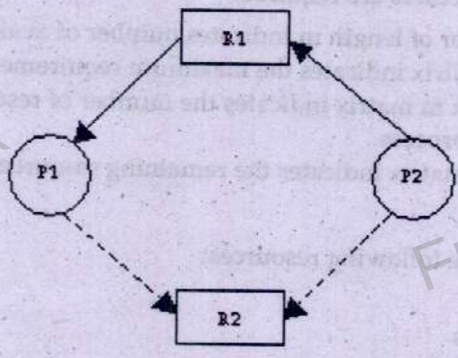


Figure: Resource-allocation graph for deadlock avoidance

Suppose, P2 requests R2. If we allocate R2 to P2, it will create a cycle in the graph and the system will enter an unsafe state. It should not be allocated to P2. The algorithm cannot be applied to a resource allocation system with multiple instances of each resource type.

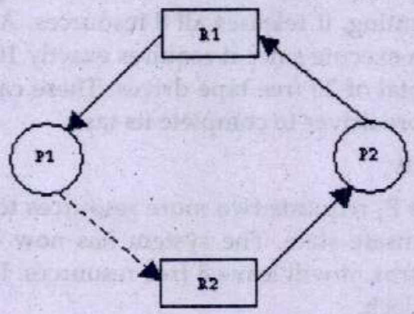


Figure: An unsafe state in resource-allocation graph

6.8.3 Banker's Algorithm

Banker's algorithm is less efficient than resource allocation graph algorithm. But it can be implemented in a system with multiple instances of each resource type. It requires that each new process should declare the maximum number of instances of each required resource type. When a process requests certain resources, the system determines whether the allocation of those resources will leave the system in safe state. If it will, the resources are allocated. Otherwise the process must wait until some other process releases the resources.

The Banker's algorithm allows the following:

- Mutual exclusion
- Wait and hold
- No preemption

It prevents the following:

- Circular wait

User process may only request one resource at a time. System grants request only if the request will result in a safe state. This algorithm uses different data structures to implement deadlock avoidance. Let **n** be the number of processes and **m** be the number of resource types. Following data structures are required:

- **Available:** A vector of length **m** indicates number of available resources of each type.
- **Max:** An **n x m** matrix indicates the maximum requirement of each process.
- **Allocated:** The **n x m** matrix indicates the number of resources of each type currently allocated to each process.
- **Need:** The **n x m** matrix indicates the remaining resource need of each process.

Example

Assume we have the following resources:

- 5 tape drives
- 2 graphic displays
- 4 printers
- 3 disks

We can create a vector representing our total resources: **Total** = (5, 2, 4, 3). Consider we have already allocated these resources among four processes as demonstrated by the following matrix named **Allocated.**

Allocated = (4, 2, 2, 3).

We also need a matrix to show the number of each resource still needed for each process. We call this matrix **Need.**

Process Name	Tape Drives	Graphics	Printers	Disk Drives
Process A	2	0	1	1
Process B	0	1	0	0
Process C	1	0	1	1
Process D	1	1	0	1

Table: Allocation matrix

Process Name	Tape Drives	Graphics	Printers	Disk Drives
Process A	1	1	0	0
Process B	0	1	1	2
Process C	3	1	0	0
Process D	0	0	1	0

Table: Need matrix

Now the vector **Available** = **(1, 0, 2, 0).**

Working of Banker's Algorithm

1. Find a row in the **Need** matrix, which is less than the **Available vector.** If such a row exists, then the process represented by that row may complete with those additional resources. If no such row exists, deadlock is possible.
2. You want to double-check that granting these resources to the process for the chosen row will result in a safe state. Suppose that the process has acquired all its needed resources, executed, terminated, and returned resources to **Available** vector. Now the value of **Available** vector should be greater than or equal to the previous value.
3. Repeat steps 1 and 2 until:

- All the processes have successfully reached pretended termination. This indicates that the initial state was safe. OR
- Deadlock is reached. This indicates that the initial state was unsafe.

Following is the working of the above algorithm:

Iteration 1:

1. Examine the **Need** matrix. The only row that is less than the **Available** vector is the one for Process D.

Need (Process D) = (0, 0, 1, 0) < (1, 0, 2, 0) = **Available**

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

2. If we assume that Process D completes, it will turn over its currently allocated resources, incrementing the **Available** vector.

(1, 0, 2, 0)	Current value of available
+ (1, 1, 0, 1)	Allocation (Process D)
(2, 1, 2, 1)	Updated value of Available

Iteration 2:

1. Examine the **Need** matrix, ignoring the row for Process D. The only row that is less than the **Available** vector is the one for Process A.

Need (Process A) = (1, 1, 0, 0) < (2, 1, 2, 1) = **Available**

2. If we assume that Process A completes, it will turn over its currently allocated resources, incrementing the **Available** vector.

(2, 1, 2, 1)	CURRENT VALUE OF AVAILABLE
+ (2, 0, 1, 1)	Allocation (Process A)
(4, 1, 3, 2)	Updated value of Available

Iteration 3:

1. Examine the **Need** matrix without the row for Process D and Process A. The only row that is less than the **Available** vector is the one for Process B.

Need (Process B) = (0, 1, 1, 2) < (4, 1, 3, 2) = **Available**

2. If we assume that Process B completes, it will turn over its currently allocated resources, incrementing the **Available** vector.

(4, 1, 3, 2)	CURRENT VALUE OF AVAILABLE
+ (0, 1, 0, 0)	Allocation (Process B)
(4, 2, 3, 2)	Updated value of Available

Iteration 4:

1. Examine the **Need** matrix without the rows for Process A, Process B, and Process D. The only row left is the one for Process C, and it is less than the **Available** vector.

Need (Process C) = (3, 1, 0, 0) < (4, 2, 3, 2) = **Available**

2. If we assume that Process C completes, it will turn over its currently allocated resources, incrementing the **Available** vector.

(4, 2, 3, 3)	CURRENT VALUE OF AVAILABLE
+ (1, 0, 1, 1)	Allocation (Process C)
(5, 2, 4, 3)	Updated value of Available

Notice that the final value of the **Available** vector is the same as the original **Total** vector, showing the total number of all resources:

Total = (5, 2, 4, 2) < (5, 2, 4, 2) = Available

This means that the initial state represented by the **Allocation** and **Need** matrices is a **safe state**. The safe sequence that assures this safe state is <D, A, B, C>.

6.9 Deadlock Detection

If the system does not use any of the preventive or avoidance methods, there is a possibility of deadlock occurrence. If a deadlock occurs in the system, it must be detected and recovered. The system needs to provide some methods of deadlock detection and recovery. These methods are overhead for the system

6.9.1 Single Instance of Each Resource Type

Suppose that each resource type has only one instance and there is only 1 printer, 1 hard drive, 1 tape drive, etc. A variant of the **resource-allocation graph** can be used called **wait-for graph**. It is done by removing the nodes of type resource and collapsing the appropriate edges.

In this graph, processes are not waiting for resources. They are waiting for other processes that hold these resources. A cycle in the graph indicates the occurrence of a deadlock in the system.

This approach requires that a cycle detection algorithm runs from time to time to detect deadlock. Algorithms that detect cycles are $O(N^2)$. N is the number of vertices in the graph.

6.9.2 Detection-Algorithm Usage

Detection algorithms need to be executed to detect a deadlock. The frequency and time when we run such algorithm is dependent on how often we assume deadlocks occur and how many processes they may effect.

If deadlocks may happen often, we run the detection often. If it affects many processes, we may decide to run it often so that less processes are affected by the deadlock.

We could run the algorithm on every resource request. The deadlocks are rare so it is not very efficient use of resources. We could run the algorithm from time to time like every hour or at random times during the system execution lifetime.

6.10 Deadlock Recovery

We can recover from a deadlock by using two approaches:

- Kill the processes that releases all resources for killed process
- Take away resources.

6.10.1 Process Termination

When recovering from a deadlock by process termination, we have two approaches.

- We can terminate all processes involved in a deadlock
- We can terminate them one by one until the deadlock disappears

Killing all processes is costly since some processes may have been doing something important for a long time and will need to be re-executed again.

Killing one process at a time until deadlock is resolved is also costly. We must run deadlock detection algorithm every time we terminate a process to make sure we got rid of the deadlock.

Some priority must be considered when terminating processes because we do not want to kill an important process when less important processes are available. Priority might also include things like

- How many resources are being held by that process
- How long has it executed
- How long it has to go before it completes
- How many processes will have to be terminated
- How many resources it needs to complete its job etc.

6.10.2 Resource Preemption

This approach takes resources from waiting processes and gives them to other processes. The victim process cannot continue regularly. We have a choice of how to handle it. We can either terminate that process or roll it back to some previous state so that it can request the resources again.

Some important factors for applying resource preemption are as follows:

1. Selection of the Victim

The most important decision is the selection of the victim process to be preempted. It will be determined by considering the number of resources held by that process, the amount of time consumed by the process etc.

2. Rollback

When a resource is preempted from a process, it cannot continue its normal execution. We should rollback that process to a safe state so that it can be restarted later.

3. Starvation

It is possible that the same process is selected as victim process for preemption. If this happens, the process will not continue and a starvation situation will occur. We should ensure that no process should be selected as a victim more than a limited number of times.

If the system has resource preemption, a deadlock cannot occur. The type of resource preemption we are talking about is abnormal preemption. It only occurs when a deadlock detection mechanism detects a deadlock.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

Review Questions

1. What is deadlock?

A set of processes are deadlocked if each process in the set is waiting for an event that only another process in the set can cause.

2. Write the four necessary conditions for a deadlock to exist.

The four necessary conditions for a deadlock to exist are mutual exclusion condition, wait-for condition, no-preemption condition and circular wait condition.

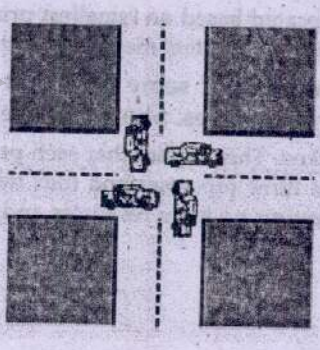
3. Write three examples of deadlocks that are not related to computer environment.

1. Two cars crossing a single-lane bridge from opposite directions.
2. A person going down a ladder while another person is climbing up the ladder.
3. Two trains traveling toward each other on the same track.

4. Is it possible to have a deadlock involving only one single process?

No. This follows directly from the hold-and-wait condition.

5. Show that the four necessary conditions for deadlock indeed hold in this example.



Each cross of the streets is considered as a resource. Each line of cars is considered as a process. The mutual exclusion exists as only one line of cars at a time can use the resource. The hold and wait exists as each line of cars is holding one resource and is waiting for the next resource. The No preemption exists as the resource cannot be released until the whole line of cars has passed it. The circular wait exists as there are 4 lines of cars L1, L2, L3, L4. L1 is waiting for L2, L2 is waiting for L3, L3 is waiting for L4, L4 is waiting for L1.

6. State a simple rule that will avoid deadlocks in this system.

There are many ways to avoid the deadlocks in this system. One way is to break the second condition. A line of cars cannot hold a cross and wait. It means that no car can stay in the cross.

7. Write the method of preventing the hold and wait condition.

The hold-and-wait condition can be prevented by requiring that a process request all of its required resources at one time. It should block process until all requests can be granted simultaneously.

8. Write the method of preventing the circular wait condition.

The circular-wait condition can be prevented by defining a linear order of resource types. Suppose that a process has been allocated resources of type R. It may subsequently request only those resources of types following R in the ordering.

9. What is difference between deadlock prevention, deadlock avoidance and deadlock detection?

Deadlock prevention restricts resource requests to prevent at least one of the four conditions of deadlock. It can be done by preventing one of the three necessary conditions of mutual exclusion, hold and wait, no preemption. It can also be done by preventing circular wait. Deadlock avoidance allows the three necessary conditions but makes sure that the deadlock point is never reached. In deadlock detection, the requested resources are granted to processes whenever possible. The operating system periodically performs an algorithm to detect the circular wait condition.

10. Write two methods of preventing the no preemptible condition.

If a process holding certain resources is denied a further request, it must release its original resources and may request them again together with the additional resources. Alternatively, if a process requests a resource currently held by another process, the operating system may preempt the second process to release its resources.

11. Why deadlock recovery is a difficult problem?

Deadlock recovery is a difficult problem because it normally requires killing one or more processes to allow another process to finish. It also requires a careful preservation of the state of an existing job that is not always possible. The preserved job may be resumed which is not very easy.

12. What is starvation? Give an example.

Starvation is a situation that occurs when the system allocates resources according to some policy such that progress is being made. However one or more processes never receive the resources they require as a result of that policy.

Example is a printer that is allocated based on "smallest print job first" in order to improve the response for small jobs. A large job on a busy system may never print and thus *starve.*

13. A system consists of four resources of same type that are shared by three processes. Each process needs at most two resources. Show that the system is deadlock-free.

Suppose the system is deadlocked. This implies that each process is holding one resource and is waiting for one more. Since there are three processes and four resources, on process must be able to obtain two resources. This process requires no more resources and therefore it will return its resources when done.

7 MEMORY MANAGEMENT

Chapter Overview

7.1 Memory Management
7.2 Basic Requirements
    7.2.1 Relocation
    7.2.2 Protection
    7.2.3 Sharing
    7.2.4 Program Translation and Address Spaces
    7.2.5 Loading and Linking
        7.2.5.1 Absolute Loading
        7.2.5.2 Re-Locatable Loading
        7.2.5.3 Dynamic Loading (Dynamic Relocation)
        7.2.5.4 Linking
        7.2.5.5 Overlaying
7.3 Contiguous & Non-Contiguous Storage Allocation
7.4 Swapping
7.5 Memory Allocation Methods
    7.5.1 Single Process Allocation Scheme/Single Absolute Partition
    7.5.2 Multiple Fixed Partition Allocation Scheme
    7.5.3 Multiple Variable Partition Allocation Scheme
        7.5.3.1 Coalescing Holes
        7.5.3.2 Storage Replacement Strategies
        7.5.3.3 Compaction
7.6 Logical vs. Physical Address Space
7.7 Paging
7.8 Page Table Structure and Hardware Support
7.9 Segmentation
    7.9.1 Difference between Paging and Segmentation
7.10 Segmentation with Paging

Review Questions

7.1 Memory Management

Computer has main memory or RAM. Various architectures enable various uses of such memory. Internally, memory could be accessed in different ways. Processes cannot run unless their code and data structures are in the RAM. It is in the main memory where instructions reside and are interpreted by the processor.

Some important issues related to the need for **memory management** include:

- Many times an operating system manages many processes (multi-programming).
- The code and data for a process must be in RAM before it could be run.
- Processes must not be able to access the code and data of other processes without permission. It means that the processes must be protected.
- Processes must be able to access and share the code and data of other processes if they have permission.
- There is usually not enough RAM to hold the code and data for all the currently running processes in RAM.

The **Memory Management Module** is the part of the operating system that must solve the above issues. In other words memory management is about sharing memory so that the largest number of processes can run in the most efficient way.

7.2 Basic Requirements

Any mechanism and policy associated with memory management should satisfy certain requirements. Different requirements are as follows:

7.2.1 Relocation

The main memory is shared among different processes in multiprogramming system. It is not possible to know where the program will be loaded in memory when it is executed. The processes are also swapped in and out of main memory to maximize the utilization of processor. If a process is swapped out and then swapped in later, it is difficult to specify that it is loaded in the same memory location. It may be relocated to a different area in memory.

The operating system needs to know the location of process control information and the execution stack. It also requires the entry point to begin execution of the program for the process. The operating system manages the memory and is responsible to bring the process in main memory. The processor also deals with the memory references within the program. The processor hardware and operating system must be able to translate memory references in the program code into actual physical memory addresses. This process of converting the logical address into physical address is known as **relocation**.

7.2.2 Protection

The main memory may have many processes being executed at one time. Each process must be protected against unwanted interference by other processes. The programs in other processes should not be able to reference the memory locations in a process to read or write without permission.

A user process normally cannot access any part of the operating system. A program in one process normally cannot branch to an instruction in another process. A program in one process cannot access the data area of another process without special arrangements. The processor must abort such instructions when executed.

The operating system cannot anticipate the memory references that will be made by a program. That is why the protection requirement of memory must be satisfied by the processor not the operating system.

7.2.3 Sharing

The memory protection must be relaxed in some cases. For example, several processes may be executing the same program such as an editor. The memory is used more efficiently if each process uses the same copy of the program code. This kind of sharing is almost always done automatically by the system software.

The processes that cooperate to solve a problem may want to access shared data structures. The processes that will participate in the sharing must declare shared memory segments explicitly. The operating system is responsible for protecting the shared memory from unauthorized access by other processes. It must also ensure that the sharing processes are restricted only to the shared portion of each other's address space.

7.2.4 Program Translation and Address Spaces

A process has a logical **address space.** It is a linear set of locations for program code, variables and stack etc. As the program executes, all or part of the logical address space is mapped to a physical address space. Physical address space is a subset of primary memory. The address space is a product of the **translation-linking-loading** process.

Translators generate re-locatable object code modules from source code. The various modules are then linked to form a single module called **load module**. It is then loaded into memory. The addresses produced by the translator must be converted to physical addresses at some time. This process is called **address binding**. The addresses can be bound at compile time, load time or run time. The most common binding time in modern systems is run time.

Different types of addresses are as follows:

Symbolic addresses	The variable names and named instructions are used in source code. The translator converts them into logical addresses.
Logical addresses	Logical addresses are assigned independent of physical locations. They may still require additional mapping operations to be bound to absolute addresses.
Relative addresses	It is a kind of logical address. The address is expressed as a location relative to some known point such as the beginning of the program. It is assigned by the translator.
Physical addresses	It is the actual location in main memory at which a particular instruction or variable is loaded. It is also called absolute address.

7.2.5 Loading and Linking

The loading is performed in the following ways:

7.2.5.1 Absolute Loading

An **absolute loader** copies a load module into memory. It makes no changes to the logical addresses assigned by the translator. It assumes that the translator or the programmer assigned the correct absolute addresses. With this approach, a load module must always occupy the same set of memory locations. This approach is virtually never used.

7.2.5.2 Re-Locatable Loading

In this loading, the re-locatable loader adds an **offset** to each logical address in the program as it copies a load module into memory. This type of loader requires the translator to generate relative addresses. It is also necessary for the program to be loaded into consecutive memory locations. The physical addresses are found by adding M to each relative address. M is the start address of the load module.

In this case, addresses are bound at load time. It is more flexible approach than compile time binding. The program can be loaded into a separate set of locations each time it runs.

Relocatable loading provides more multiprogramming than absolute loading. With absolute loading, two programs that specify overlapping addresses cannot execute at the same time. The relocatable loading does not have this problem. It also also makes the programming easier. The programmer does not need to specify where a program must be located in memory.

7.2.5.3 Dynamic Loading (Dynamic Relocation)

Re-locatable loading does not support swapping. Once a process has been loaded and relocated, its addresses are absolute. A swapped-out process can only be swapped back to the same memory locations. This is very restrictive.

Dynamic loading or dynamic relocation defers the process of determining absolute addresses until the address is used. The loader places the load module into memory without adjusting relative addresses. A special register is loaded with the start address of the load module. Hardware automatically adds the register contents to a relative address each time the address is used. In this case, address binding is at run-time. Relocation is done each time the address is used. It must be handled in hardware.

7.2.5.4 Linking

A linker joins several object modules into a single load module. Input to the linker is a set of object modules. Each module has been translated with relative addresses, relative to 0 as the start address of the module. References from one module to another such as function calls, data references are still symbolic. The loader must generate a single contiguous module in which all external references have been resolved. The term **Linkage editor** is applied to this type of linker.

A dynamic linker postpones some of the linkage functions until run time. Dynamic linking can be done at load-time or at run time. Load-time dynamic linking prepares a re-locatable load module in normal way. It leaves some external references unresolved. These references are usually to the system utilities or language libraries. The system copies of the target modules can be linked at load time. The utilities can be changed without forcing programmers to re-link existing load modules. A shared code can be linked to more than one program. Run-time dynamic linking goes a step further. Some modules will not be linked in until they are actually called.

7.2.5.5 Overlaying

Overlaying is a technique that is used to execute a process even if the memory is not enough. The programmer can define two or more overlays. These overlays can execute in the memory independently. The operating system can swap overlays and manage the memory.

The overlays are useful when the program to be executed is larger than available memory. It can also be useful when the program can be broken into phases that are not required to be in main memory at the same time. The disadvantage of overlaying is that it requires extensive involvement of programmer. The programmer has to identify and define overlays efficiently. It is a very difficult task.

7.3 Contiguous & Non-Contiguous Storage Allocation

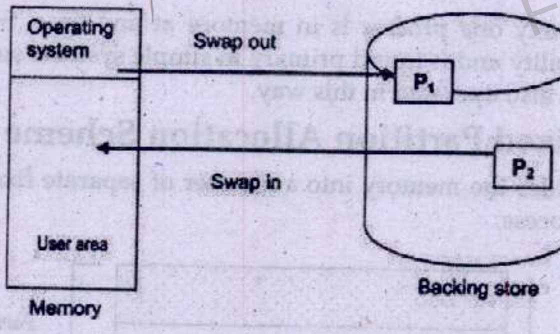
The earliest computing system required contiguous storage allocation in which each program occupied a single contiguous memory block. In these systems, the technique of multiprogramming was not possible.

In non-contiguous storage allocation, a program is divided into several blocks that may be placed in different parts of main memory. It is more difficult for an operating system to control non-contiguous storage allocation. The benefit is that if main memory has many small holes available instead of a single large hole, then operating system can often load and execute a program that would otherwise need to wait.

7.4 Swapping

A process cannot be executed in the CPU unless it is completely or partly in memory. A process not executing in CPU can be completely **swapped out** of memory to a **backing store** such as a high-speed disk. It can then be **swapped in** to memory when it needs to be executed. **Swapping** is a memory management technique that swaps a process into main memory to execute it and then swaps it out to the backing storage when required.

The swapping is done by **swapper**. Suppose that a **dispatcher** tries to load a process into CPU and finds that it is not currently in memory. It will call the swapper to swap the process back into memory.



Benefits of Swapping

Some benefits of using swapping are as follows:

- It allows higher degree of multiprogramming.
- It allows dynamic relocation. It means that the process can be swapped in different location if address binding is performed at execution time.
- It provides better memory utilization.
- It can be applied on priority-based scheduling algorithms to improve performance.

7.5 Memory Allocation Methods

The program code has to be transferred from secondary storage to main memory in order to execute a process. This activity is called **process loading**.

In some systems, the reverse operation is also used. This operation transfers a process from memory to secondary storage. This is done to accommodate and execute another process. The overall effect is the a swap of two processes. This activity is called **swapping**.

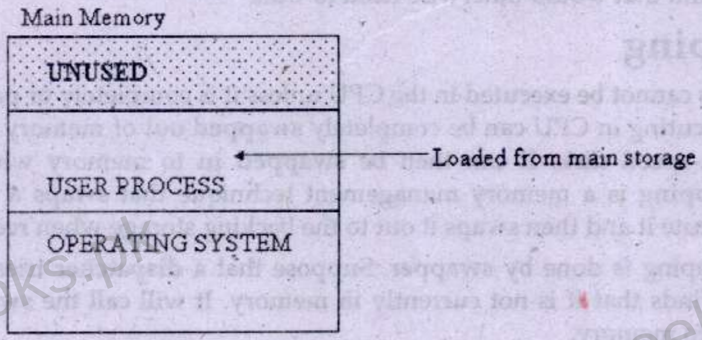
The three main memory allocation schemes are as follows:

1. **Single process allocation scheme.**
2. **Fixed partition allocation scheme.**
3. **Variable partition allocation scheme.**

In general the main memory is usually partitioned in two main parts. One is reserved for the operating system and the other to hold one or more user process.

7.5.1 Single Process Allocation Scheme/Single Absolute Partition

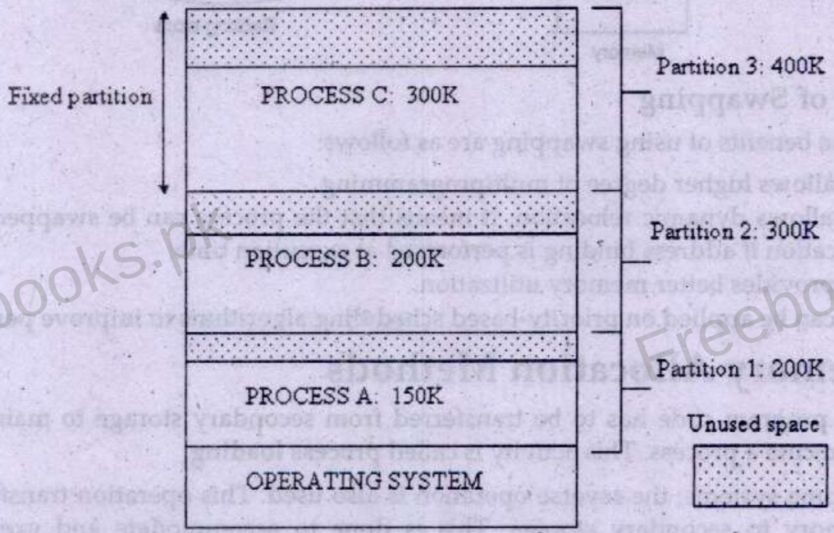
Memory management is simple in a computer that only runs one process at a time. The process to be executed is loaded into the free space area of the memory. In general, the remaining part of the memory space will not be used.


Main Memory
UNUSED
Loaded from main storage
USER PROCESS
OPERATING SYSTEM

In this scheme, only one process is in memory at any time. Such an arrangement is clearly limited in capability and is found primary in simple systems such as computer games. Early MS-DOS systems also operated in this way.

7.5.2 Multiple Fixed Partition Allocation Scheme

This scheme divides the memory into a number of separate fixed areas. Each memory area could hold one process:


Fixed partition
PROCESS C: 300K
Partition 3: 400K
PROCESS B: 200K
Partition 2: 300K
PROCESS A: 150K
Partition 1: 200K
OPERATING SYSTEM
Unused space

In the above example, the memory consists of three areas of size 200K, 300K, and 400K respectively. Each area holds a process. The number of partitions and their sizes in a practical operating system would be controlled depending on the amount of free memory available and the size of processes to be run.

For example, one partition should be large enough to hold the largest process to be run. As shown above, each partition will typically contain unused space so that the total unused space could be considerable.

The occurrence of wasted space in this way is called **internal fragmentation**. It is called 'internal' because it occurs within the space (partition) allocated to the process in question.

Typically, the partitions would be set up with a range of partitions sizes so that a mixture of large and small processes could be accommodated. This scheme was one of the first schemes to be employed in multiprogramming systems.

Advantages

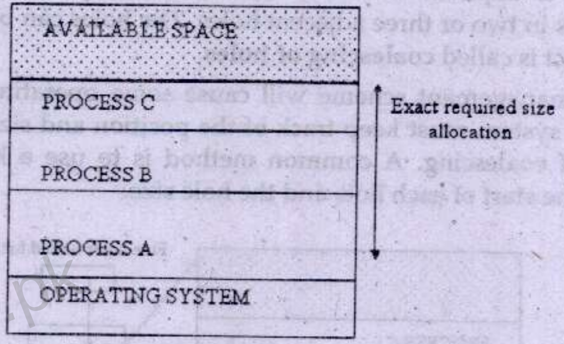
- It is relatively simple to implement
- It facilitates memory protection mechanism due to its fixed partitions.

Disadvantages

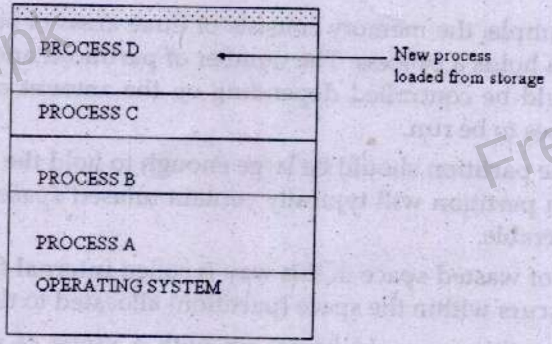
- The fixed partition sizes can prevent a process from being run due to the unavailability of a partition of sufficient size.
- Internal fragmentation wastes space that collectively could accommodate another process.

7.5.3 Multiple Variable Partition Allocation Scheme

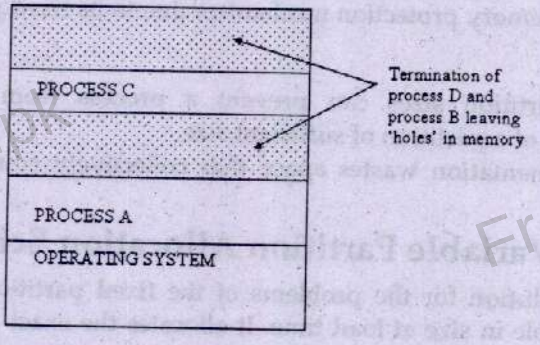
The obvious solution for the problems of the fixed partition scheme is to allow the partitions to be variable in size at load time. It allocates the exact amount of memory to the process as required. Processes are loaded into consecutive areas until the memory is filled or the remaining space is too small to accommodate another process.



If a new process D is loaded then it is allocated adjacent to process C if available space is sufficient. When a process terminates, the space it occupies is freed and becomes available for the loading of a new process. However, this reveals a problem in the nature of the technique. As processes terminate and space is freed, the free space appears as a series of 'holes' between the active memory areas.


PROCESS D
New process loaded from storage
PROCESS C
PROCESS B
PROCESS A
OPERATING SYSTEM

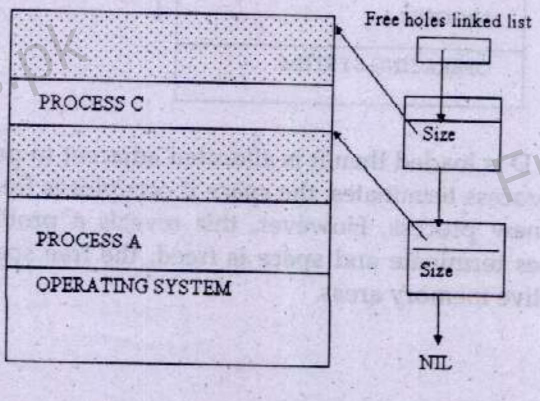
The operating system must attempt to load an incoming process into a space large enough to accommodate it. It is possible that a new process cannot be started because none of the holes is large enough even though the total free space is more than the required size. Distribution of free memory in this way is called **external fragmentation.**


PROCESS C
Termination of process D and process B leaving 'holes' in memory
PROCESS A
OPERATING SYSTEM

7.5.3.1 Coalescing Holes

A process that is adjacent to one or more holes may terminate and free its space allocation. This results in two or three adjacent holes. The holes can be viewed and utilized as a single hole. The effect is called **coalescing of holes.**

Any memory management scheme will cause some operating overhead. In variable partition scheme, the system must keep track of the position and size of each hole. It should consider the effect of coalescing. A common method is to use a linked list. A linked list contains pointers to the start of each hole and the hole size.


Free holes linked list
PROCESS C
Size
PROCESS A
Size
OPERATING SYSTEM
NIL

7.5.3.2 Storage Replacement Strategies

When a new process has to be loaded using a variable partition scheme, it is necessary to try and select the 'best' hole in which to place it. In other words how are we going to satisfy a request for an allocation of size n from a list of free holes?

A number of policies have been devised and are described below:

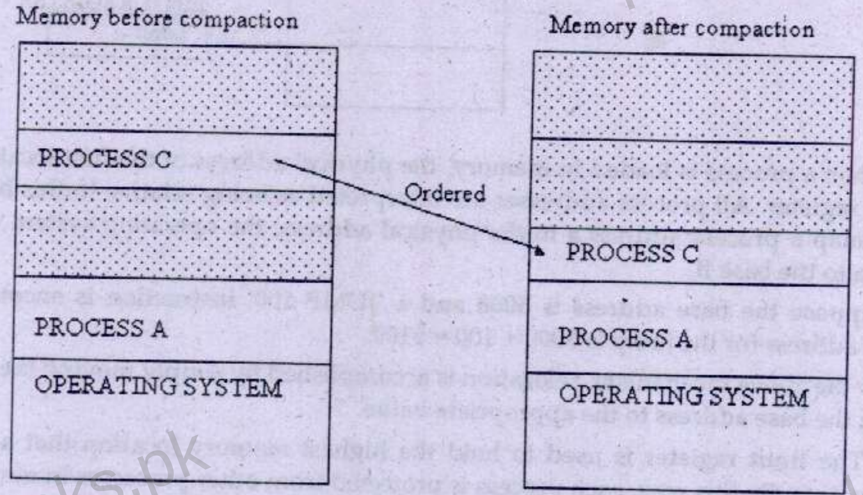
First Fit:	The incoming process is placed in the main memory in the first available hole that is big enough to hold the process.
Best Fit	The incoming process is placed in the main memory in the smallest hole that is big enough to hold the process. In this policy, the system must search entire list unless ordered by size. It produces the smallest leftover hole.
Worst Fit	The incoming process is placed in main memory in the largest hole. The system must also search the entire list. It produces the largest leftover hole.

In practice, the best fit and the worst fit generally prove to be the most effective.

The variable partition scheme has been used successfully in many systems and is a significant improvement on the fixed partition scheme. Its principal defect is the problem of fragmentation. It reduces the effective capacity of memory.

7.5.3.3 Compaction

The fragmentation problem encountered in the previous method can be resolved by physically moving the processes about the memory in order to close up the holes. It brings the free space into a single large block. This method is called **compaction.**



Compaction makes the total free space more usable for incoming processes. But an overhead is required to re-shuffle current processes. All processes must be suspended while the re-shuffle takes place. Such activity is not be feasible in a real-time system. Moreover, the operating system must decide at what interval of time compaction will be carried out.

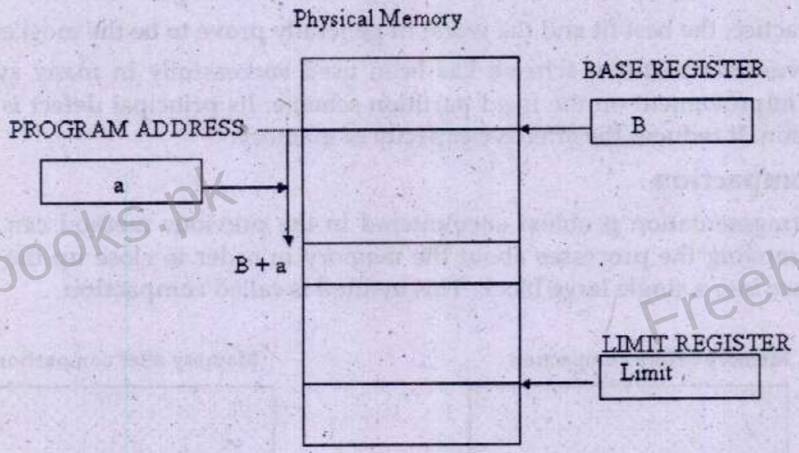
In practice, the compaction scheme is usually not used. Its overheads and added complexity minimize its advantage over non-compacted schemes.

7.6 Logical vs. Physical Address Space

A process can change place in memory for the following three main reasons:

- A process must be unloaded from memory if its CPU time-slice has ended. It is placed in the ready queue to be loaded again later. It allows the memory space for another process to be loaded.
- A must be unloaded from memory if it enters a block state by calling I/O function etc. It is then placed in the blocked queue to be loaded later.
- The process may change its position if compaction is applied.

In all the above cases, the process is being reallocated in memory. It must be possible for a process to be loaded at any address selected by the operating system. The mechanism involved in the reallocation of processes is shown below:



When a process is loaded in memory, the physical address of the first location is put in the **base register**. All process addresses are interpreted as being relative to this base address. Thus to map a process address a to the physical address, the operating system will add the address a to the base **B**.

Suppose the base address is 5000 and a 'JUMP 100' instruction is encountered. The physical address for the jump is 5000 + 100 = 5100.

By the above mechanism, relocation is accomplished by simply moving the process and resetting the base address to the appropriate value.

- The **limit register** is used to hold the highest memory location that a process can access. By this way, each process is protected from other processes in memory.
- A **logical address** is expressed as a location relative to the beginning of the process (program). Instructions in the program contain only logical addresses.
- A **physical address** is an actual location in main memory.

7.7 Paging

Physical memory is divided into fixed-size blocks called **frames**. Logical memory is also divided into the same size blocks called **pages**. A single process may require many pages of memory. When a process is scheduled for execution, its pages are loaded into any available memory frames from the backing store.

The backing store is also divided into blocks. The blocks are of the same size as the frames in main memory. Paging requires additional hardware support in the form of a page table. Every logical address generated by the CPU consists of two parts:

- A page number (p)
- A page displacement (d).

The page number is used as an index into the page table. The page table contains the base address of each page in physical memory. This base address is combined with the page displacement to determine the physical memory address. It is then sent to the memory management unit. It completes the translation from logical address to physical address.

The page size and frame size are hardware dependent. It is common that both will contain a number of bytes that is a power of 2. The typical values will range from 512 bytes to 16 MB per page. An integral size based upon a power of two is used due to the ease of translation that can occur with such numbers.

Suppose the logical address space consists of 2^m bytes and the page/frame size is 2^n bytes. The high-order *m-n* bits of a logical address will designate the page number. The low order *n* bits will designate the page displacement as shown below.

Page number **p**	Page displacement **d**
m-n bits	n bits

Example

Suppose a system has a page size of 8 bytes ($2^3 = 8$). The physical memory consists of 1024 bytes. It is divided into 128 frames. Suppose that the logical address space consists of 128 bytes ($2^7 = 128$). Thus, m = 7 and n = 3 so the page number will require m-n = 7-3 = 4 bits. The displacement within the page will require n = 3 bits.

The 128 byte logical address space will be divided into a total of 16 pages. It requires a total of 16 frames of physical memory. To distinguish uniquely, 16 possible frames will require 4 bits. A total of 8 bytes of code from the process will be located within each frame. It will require a total of 3 bits to uniquely determine which byte within the page/frame is being referenced by the CPU.

Suppose the CPU generates a logical address of 64. The address is at the mid-point of the process requiring 128 bytes of memory. The physical address corresponding to this logical address will be in page number 8 with displacement 0.

Page	Displacement							
	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63
8	64	65	66	67	68	69	70	71

9	72	73	74	75	76	77	78	79
10	80	81	82	83	84	85	86	87
11	88	89	90	91	92	93	94	95
12	96	97	98	99	100	101	102	103
13	104	105	106	107	108	109	110	111
14	112	113	114	115	116	117	118	119
15	120	121	122	123	124	125	126	127

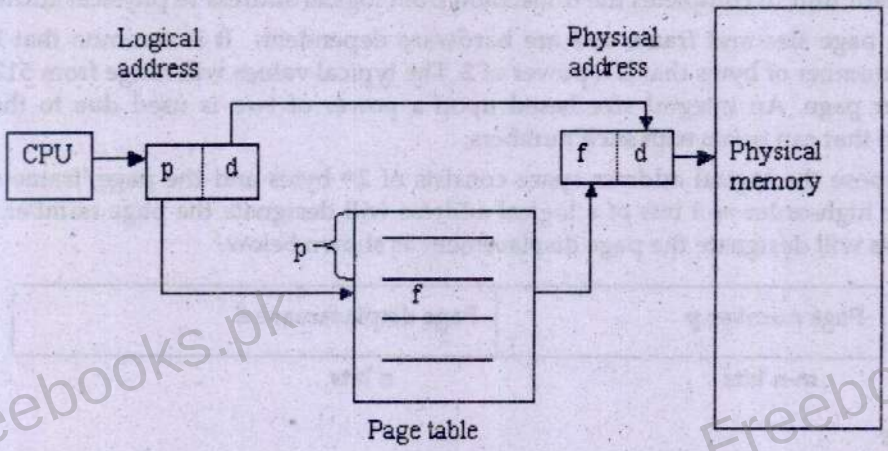
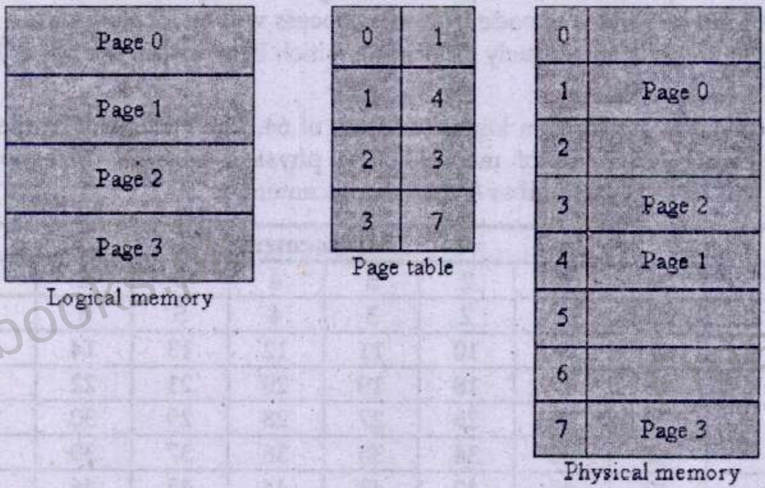

Logical address
Physical address
CPU
p
d
f
d
Physical memory
p
f
Page table

Figure: Hardware support for paging allocation scheme

The page table contains the "mapping" of the logical pages to the physical frames. This is shown graphically in the next diagram.


Page 0
Page 1
Page 2
Page 3
Logical memory
0
1
1
4
2
3
3
7
Page table
0
1
Page 0
2
3
Page 2
4
Page 1
5
6
7
Page 3
Physical memory

Using the previous diagram as a reference, assume that each page contains 4 bytes. This means that the logical address space consists of 16 bytes. Page 0 contains logical addresses 0, 1, 2, and 3. Page 1 contains logical addresses 4, 5, 6, and 7 and so on.

In the physical memory, frame number 0 holds physical address values 0, 1, 2, and 3, frame number 1 holds physical addresses 4, 5, 6, and 7, and so on.

Logical address				
Page	Displacement			
	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

Logical memory

0	1
1	4
2	3
3	7

Page table

Frame	Address			
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15
4	16	17	18	19
5	20	21	22	23
6	24	25	26	27
7	28	29	30	31

Physical memory

The page table shows that:

- Logical page 0 is in physical frame 1
- Logical page 1 is in physical frame 4
- Logical page 2 is in physical frame 3
- Logical page 3 is in physical frame 7.

To find logical address 10 in the physical memory, we require to find the physical frame that currently holds logical page 2 since logical address 10 is in logical page 2.

The page table currently shows logical page 2 to be in physical frame 3. Logical address 10 is at displacement 2 in page 2. Physical frame 3 and displacement 2 represents physical address 14. This is the address in the physical memory that currently holds logical address 10. The translation of the logical address to the physical address follows the form:

Physical address = (frame# × number bytes/page) + displacement

For this example, we have: $(3 \times 4) + 2 = 14$

Similarly, logical address 7 is page 1 displacement 3 [(1,3)]. The page table shows page 1 in frame number 4 currently so:

$(4 \times 4) + 3 = 19.$

7.8 Page Table Structure and Hardware Support

Different operating systems have different methods for implementing page tables. Most allocate a page table per process. A pointer to page table of each process is maintained with all of the other register values associated with the process and stored in its PCB.

Reloading these register values and defining the correct hardware page table values from the stored process/user page table is part of a context-switch.

The actual implementation of the page table can be accomplished in several different ways. The simplest technique involves defining a dedicated set of registers to be used as the page table. Since every memory access requires going through the page table, these registers need very high-speed logic associated with them to make the address translation efficient. This register technique works reasonably well if the size of page table is relatively small.

Most modern systems allow extremely large page tables i.e. a million entries or more. For such machines, the register implementation of the page table is not feasible primarily due to cost considerations. Instead of registers such machines keep the page table in main memory and use a **page table base register (PTBR)** to maintain a pointer to the page table. Changing page tables during a context switch requires 'changing only the value in this register rather than physically loading a large number of registers.

This approach has an inherent problem with the time that is required to access a logical memory address. To access logical address n, access must first be into the page table using the **PTBR** offset by the logical page number in which n is located. This requires a memory access. This access provides the physical frame number in which the logical page holding n is currently located. Then the memory access to this address occurs. This scheme requires two memory accesses for every logical address generated by the CPU. This doubles the time required to perform a physical memory access when the CPU generates a logical address. Such an increase in the time required to perform a memory access is not tolerable.

The standard solution to this problem is to use **content-associative memory (CAM)**. It is also called **content-addressable memory, associative registers** or **translation look-aside buffers (TLBs)**. CAM is built from extremely high-speed memory where each cell (these can be thought of as registers) consists of two parts: a key and a value.

When the CAM is presented with an item to match, that item is compared with all the keys simultaneously and if one of the cells keys matches with the item to be matched, its value component is output.

When used as a page table, the CAM is presented with an item to match that represents a logical page number. Each cell in the CAM represents one page table entry where the value part of the cell holds the physical frame number in which the logical page currently resides. This is the value, which is output by the cell. If the logical page number is found in the CAM, its frame number becomes immediately available and is used to access the physical memory. While this type of memory is quite expensive, it is also extremely fast. Typically a CAM used for such purposes contains between 8 and 2048 cells. If the page number is not in the CAM, then a memory reference to the page table (in memory) must be made. When the frame number is obtained from the physical memory then it can be used for the translation and performing the second memory access. This page number and frame number are added to the CAM so that on the *next* request it will be in the CAM.

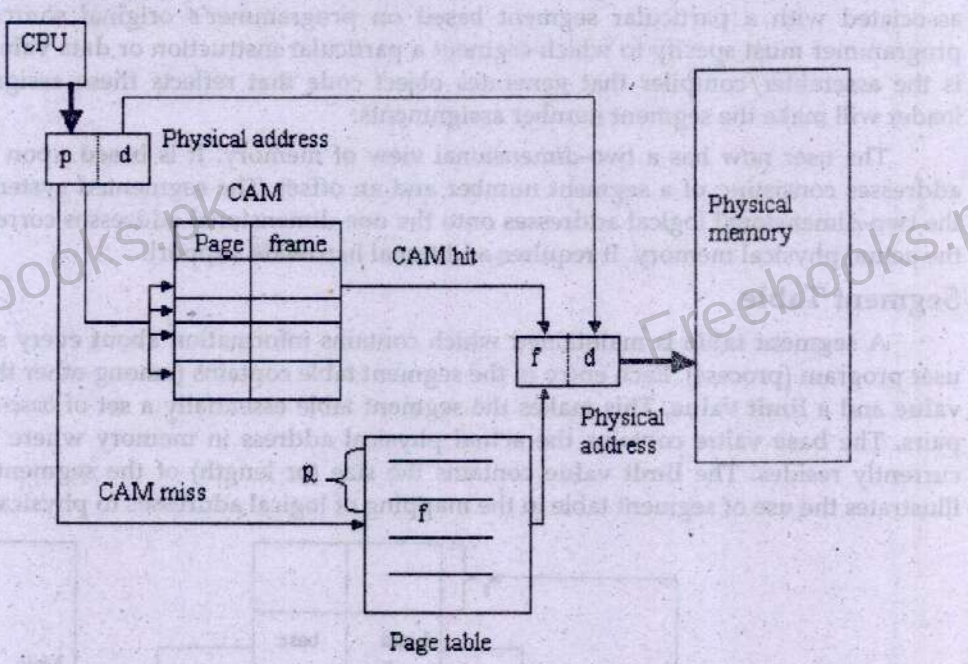
If the CAM is already full, the operating system must select a CAM entry for removal so that the one can be entered. Operating system will use a CAM entry replacement protocol for the basis of this decision. Each context-switch will require that the CAM be flushed to ensure that the next process does not use the translation information left behind by the process just switched out.

Each logical address request generated by the CPU whose translation information is in the CAM at the time of the request is called a CAM **hit**. The percentage of time that this occurs is called the CAM **hit ratio**. An 86% hit ratio means that 86% of the time the necessary translation information is in the CAM. For example, if it takes 15 nanoseconds to search the CAM and a memory access requires 100 nanoseconds, then the mapped memory access requires a total of 115 nanoseconds if there is a CAM hit. If there is a CAM miss on the logical address request, then the total time required will be 215 nanoseconds, since two memory accesses are required in addition to the CAM search time. Assume negligible time to add the new entry to the CAM – although in reality this time is not negligible. To find the *effective memory access time* (its like the average access time under these conditions) each case must be multiplied by its probability, which gives:

$$\textit{Effective memory access time} = (0.86 \times 115) + (0.14 \times 215) = 98.9 + 30.1 = 129$$

Thus, the effective memory access time is 129 nanoseconds that represent a 14% slow down approximately.

The hit ratio is related to the number of cells in the CAM. When the number of cells ranges between 16 and 512 a hit ratio of between 80%-98% can be achieved. Intel's 80486 chip uses 32 cells. Following diagram shows the address translation that occurs when using a CAM to speed-up page table look-up.



7.9 Segmentation

Segmentation is a process that breaks a program into logical segments and allocates the space for these segments in memory separately. The segments can be of variable lengths. They may not be allocated contiguous memory. The segments are more flexible. They are visible to the programmers. The programmer can use them to tell the system to bring certain data into main memory always together. The disadvantage of segmentation is that the programmer must manage segments whether he wants or not. Segmentation suffers from external fragmentation that is more difficult to deal.

The systems utilizing a paged memory management system must handle the mapping of logical memory onto the physical memory. In a paged environment, the user's view of memory (the logical view) is much different than system's view of memory. Typically, the user views memory as a collection of different sized segments containing different components of their application with no implied ordering amongst the segments.

For example, the user will view one segment as containing their main program. He views another segment containing a subroutine. Another segment may contain common library routines. Another segment holds the symbol table for their program. A user does not view the memory as a contiguous sequence of bytes. Items in a segment are addressed by their offset within the segment such as "the third symbol in the symbol table" or "the 15th line of code".

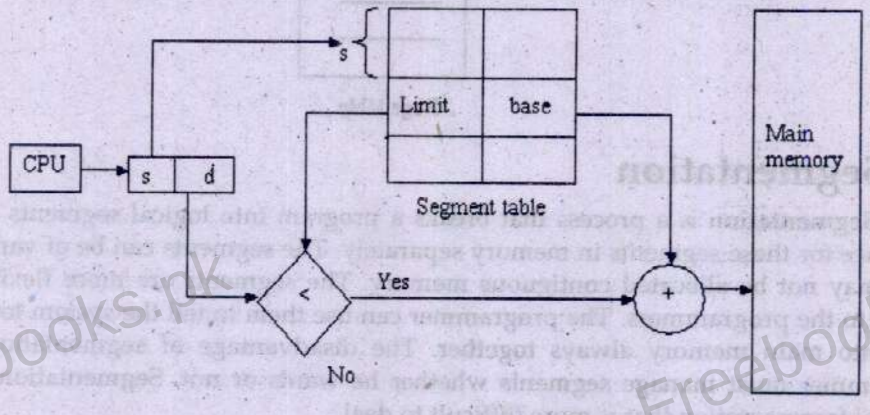
The segmentation views the logical address space as a collection of segments where each segment has a name (typically just an integer number) and a length. Logical addresses are generated by the programmer and transmitted to MMU by CPU. They must specify both the segment name and the offset in the segment.

The paging system partitioned a single logical address into **page number** and **offset** components. A logical address in a system using segmentation consists of <segment, offset>. The assembler or compiler generates object code in which each line of object code is associated with a particular segment based on programmer's original source code. The programmer must specify to which segment a particular instruction or data value belongs. It is the assembler/compiler that generates object code that reflects these assignments. The loader will make the segment number assignments.

The user now has a two-dimensional view of memory. It is based upon their logical addresses consisting of a segment number and an offset. The segmented system must map the two-dimensional logical addresses onto the one-dimensional addresses corresponding to the actual physical memory. It requires additional hardware support.

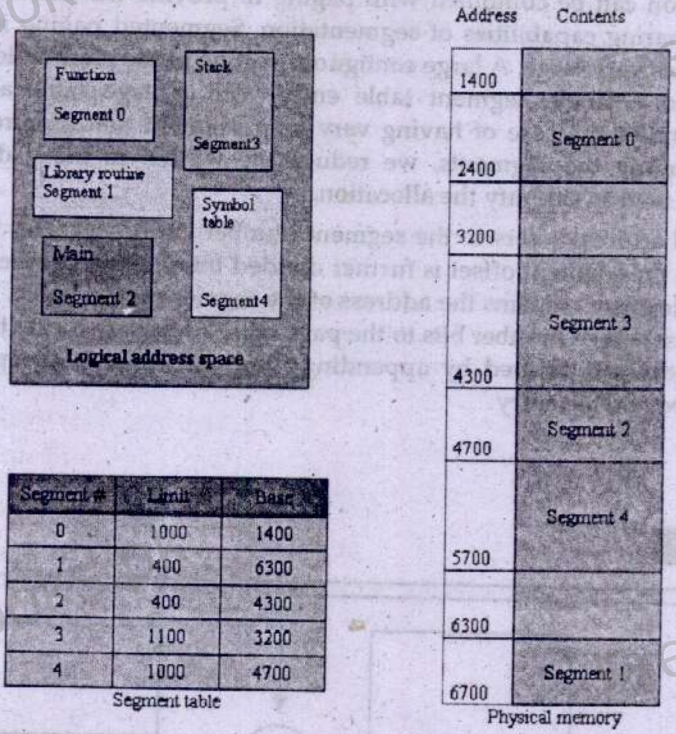
Segment Table

A **segment table** is maintained which contains information about every segment in a user program (process). Each entry in the segment table contains (among other things) a **base value** and a **limit value.** This makes the segment table essentially a set of *base-limit* register pairs. The **base value** contains the actual physical address in memory where the segment currently resides. The **limit value** contains the size (or length) of the segment. The figure illustrates the use of segment table in the mapping of logical addresses to physical addresses.



Using segmentation, the segment number is used as an index into the segment table and the base address for that segment is returned. The offset of the logical address must be between 0 and the segment limit. If it is not, then this error will be trapped to operating system. If the offset value is within the limit as specified by the segment table, then this value (the offset) is added to the base value for the segment. It will produce the address in the physical memory corresponding to the desired byte.

When the CPU generates a request for logical address <3, 649> (segment 3, offset 649), the segmentation hardware will convert as follows:


Function
Segment 0
Stack
Segment3
Library routine
Segment 1
Symbol table
Main
Segment 2
Segment4
Logical address space
Address
Contents
1400
Segment 0
2400
3200
Segment 3
4300
Segment 2
4700
Segment 4
5700
6300
Segment 1
6700

Segment #	Limit	Base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

Segment table

Physical memory

The segment table has segment number 3 with a base address of 3200 and limit of 1100. Since 649 is within the limit, the physical address is calculated as 3200 + 649 = 3849. A request for logical address <3, 1250> causes an exception since segment 3 has a limit value of 1100 and the specified offset is beyond the end of this segment and is thus an addressing violation.

7.9.1 Difference between Paging and Segmentation

The difference between paging and segmentation is as follows:

Paging	Segmentation
1. The pages do not vary in size.	1. The segments may vary in size.
2. A program is broken into pages by compiler or memory management system.	2. The program segments are specified by the programmer to the compiler.
3. In paging, the internal fragmentation is within frames.	3. In segmentation, there is no internal fragmentation.
4. There is no external fragmentation in paging.	4. Segmentation suffers has the external fragmentation.
5. The operating system must maintain a page table for each process to show the frames occupied by each page.	5. The operating system must maintain a segment table for each process to show the load address and length of segment.
6. The operating system must maintain a free frame list.	6. The operating system must maintain a list of free holes in main memory.
7. The processor uses page numbers and offset to calculate absolute address.	7. The processor uses segment number and offset to calculate absolute address.

7.10 Segmentation with Paging

Segmentation can be combined with paging to provide the efficiency of paging with protection and sharing capabilities of segmentation. Segmented paging is helpful when the page table becomes very large. A large contiguous section of the page table that is unused can be collapsed into a single segment table entry with a page table address of 0. Page segmentation handles the case of having very long segment that requires a lot of time for allocation. By paging the segments, we reduce the wasted memory due to the external fragmentation as well as simplify the allocation.

The logical address refers to the segment number and offset with the segment. When paging is added, the segment offset is further divided into a page number and a page offset. The segment table entry contains the address of the segment's page table. The hardware adds the logical address's page number bits to the page table address to locate the page table entry. The physical address is formed by appending the page offset to the page frame number specified in the page table entry.

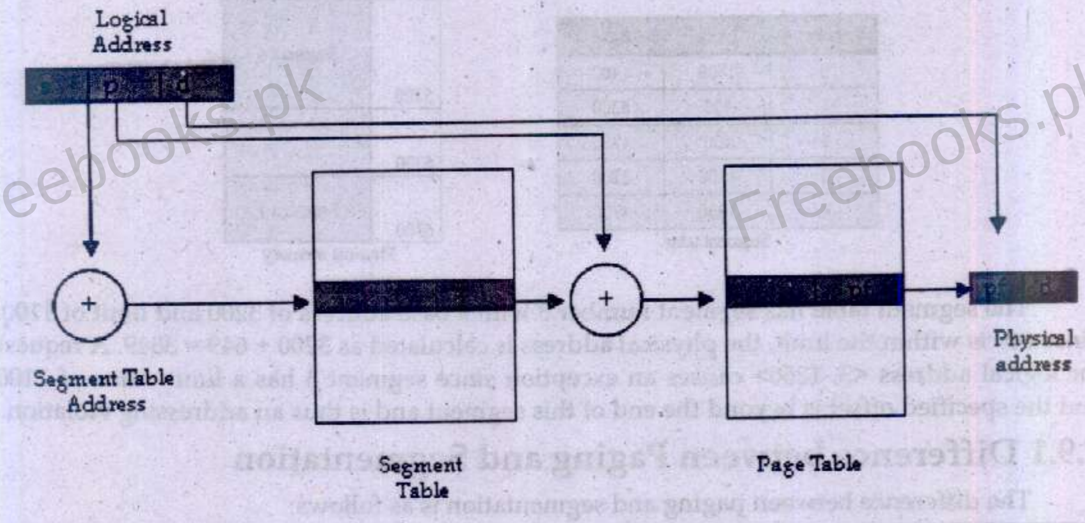


Figure: Segmentation with Paging

Review Questions

1. What are the functions of memory management? Why is it necessary?

(a) Manage one of the main resources of the machine: keep track of what parts are in use and by whom; allocate memory to processes that need it.

(b) Allow several processes to share memory by protecting it against unauthorized access.

(c) Manage the automatic movement of data between main memory and disk.

2. What is virtual address and real address?

A virtual address is a memory location in virtual memory. This location can be on disk and it can be in main memory at some times. A real address is an address in main memory.

3. Differentiate between logical, physical and relative address.

A logical address is a reference to a memory location independent of the current assignment of data to memory. A translation must be made to a physical address before the memory can be accessed. A relative address is an example of logical address in which the address is expressed as a location relative to some known point such as the beginning of the program. A physical address or absolute address is an actual location in main memory.

4. Why should the processor have the capability to relocate?

It is not possible for the programmer to know in advance which other programs will run in main memory at the time of execution of a program. The processes are also swapped in and out of main memory to maximize processor utilization. In both cases, the specific location of the process in main memory is unpredictable. So the processor must be able to relocate to meet this situation.

5. What is dynamic loading?

It is the process of loading a routine only when it is called and not yet in memory. The routine will not be loaded into memory until first time it is called.

6. What is the advantage of dynamic loading?

Routines that are never used are never loaded. More free memory is available.

7. Why memory protection is not possible at compile time?

Memory protection is not possible at compile time because the location of a program in main memory cannot be determined. It is impossible to check absolute addresses at compile time to assure the protection. Most programming languages allow the dynamic calculation of addresses at run time. All memory references generated by a process must be checked at run time to ensure that they refer only to the memory space allocated to that process.

8. What is the importance of relocatable loading?

Relocatable loading allows more multiprogramming on a system than absolute loading. In absolute loading, two programs that specify overlapping addresses cannot execute at the same time. The relocatable loading does not have this problem. The programmer does not need to specify the location in memory for loading the program. It makes the programming easier.

9. Differentiate between absolute loaders and relocating loaders.

An absolute loader places programs and data directly at the memory addresses specified by the linker or programmer. A relocating loader converts relative addresses from the load module into absolute addresses determined by the offset of an address within a memory block. Absolute loaders are simpler to build and execute faster than relocating loaders. Relocating loaders are more flexible and they minimize waste of storage.

10. What is meant by term "address binding"? When can address binding take place?

Address binding is the mapping of an address from one address space to another. Address binding can take place at:

- Compile time: Binding symbolic addresses to re-locatable addresses or absolute addresses
- Load time: Binding re-locatable addresses to absolute addresses
- Execution time: Re-locatable addresses are bound to absolute address by MMU as needed.

11. When are the overlays useful?

The overlays are useful when the program to be executed is larger than available memory. It can also be useful when the program can be broken into phases that are not required to be in main memory at the same time.

12. What is the memory management unit (MMU) and what does it do?

MMU is a device that performs the run-time mapping of logical addresses to physical addresses.

13. What is dynamic linking?

In dynamic linking, linking of library code is delayed until a routine from that library is called by the process. Instead of linking the code a stub is included in the process image for each library-routine reference. The stub is a small piece of code that indicates how to locate the appropriate memory-resident library routine or how to load the library if the routine is not present.

14. Which technique can be used to load programs into memory in any location?

The code can be written in a way that all references to global variables/procedures/functions are done via an offset from PC. It is position independent code. A bit for every word of the program can be stored whether this word contains an address or not. These bits can be used at load time to relocate the words that contain addresses. Every memory reference can be translated by an MMU via base & limit registers, paging etc.

15. Where is MMU in computers? Which memory references do not go through MMU?

MMU exists between CPU and the bus, usually next to (or on) the CPU chip. Memory references that are issued on behalf of MMU (lookups of the page table) should not go through MMU. Otherwise an infinite loop may result.

16. What is a relocation register?

A base register used to give the lowest physical address for a process.

17. What is swapping?

Copying a process from memory to disk to allow space for other processes.

18. List ways of reducing the context switch time.

a. Minimize the amount of memory of a process to be swapped.

b. Increase the speed of the disk used for swapped-out processes.

c. Overlap swapping and program execution.

19. List three ways to treat jobs, started in a given partition, that request too much memory.

a. Abort job with run-time message.

b. Return control to job. If it cannot adjust to smaller memory then abort.

c. Swap job out.

20. What is the system manager's main problem in using fixed partitions?

Determining the optimum number of partitions and their size.

21. Discuss the advantages and disadvantages of noncontiguous memory allocation.

The main advantage is that jobs can run even if their memory requirement exceeds the size of the largest available free memory area.The disadvantage is the increased cost and complexity of the hardware and the operating system.

22. Consider the following memory map:

A	200K used
B	300K unused
C	100K used
D	200K used
E	100K unused
F	300K used

Discuss alternatives to compact this memory, specifically the overhead required for compaction. Would it change anything if the sizes of some regions vary slightly from multiples of 100K?

23. Answer:

(1) To compact everything towards the top, move C & D to B and F to C & D. Total: 600K.

(2) Can do (1) smarter: move F to B & C. Total: 300K.

(3) To compact everything towards the bottom, move C & D to D & E. and A to B & C. Total: 500K.

(4) Can do (3) smarter: move C to E, and A to B & C. Total: 300K.

(5) To compact towards both edges, move C & D to B. Total: 300K.

(6) Can do (5) smarter: move C to E. Total: 100K.

In general, it can be very difficult to find the optimal compaction sequence. If hole E was a little bit larger than process C, (6) would not compact all holes into one. But it would still be a good thing to do. If process C were a little bit larger than hole E, (6) would not be viable at all. If we are willing to lose small amounts of memory, the time required for the compaction may be greatly reduced.

24. List the common allocation methods. Which is the poorest?

a. First-fit. b. Best-fit. c. Worst fit – poorest.

25. What are variable partitions?

Partitions that can be moved in location and can be changed in number.

26. What is compaction? Why use it?

Movement of processes to eliminate small partitions (holes). It allows smaller partitions to form fewer larger ones to allow larger processes to run.

27. What is paging?

Splitting program into a group of equal-sized partitions to allow the parts to be non-contiguous.

28. What is segmentation?

It is process of breaking program into logical segments and allocating space for these segments in memory separately. The segments may be of variable length and need not be allocated contiguously.

29. What is a frame?

Frame is a fixed-size block of physical memory. Each block is of same size as page.

30. Differentiate between segmentation & paging? List the advantages & disadvantages.

The main differences are:

(a) Segments are visible to the programmer but pages are not

(b) Segments vary in size but pages do not.

(c) Segments have separate protection but pages have no separate protection.

(d) Segments share code but pages do not share code

The advantages of segments are that they can be more flexible. The programmer can use them to tell the system to bring certain data into main memory always together.

The disadvantages of segmentation are that programmers must manage the segments whether they want or not. It suffers from external fragmentation that is much harder to deal than paging's internal fragmentation. Unlike paged systems, segmented systems need an address adder.

31. List advantages and disadvantages of contiguous memory allocation.

Advantage: Easy translation from logical to physical addressing. Only requires two registers i.e. limit register and relocation register.

Disadvantage: Need to find the hole big enough to fit. Leads to external fragmentation.

32. List advantages and disadvantages of paging.

Advantage: No need to find a contiguous space of the right size. Can allocate any free page. No external fragmentation as page/frame sizes are fixed. Makes sharing of memory easier.

Disadvantage: Internal fragmentation - on average half a page wasted per process. Translation of logical addresses to physical addresses is more complex. It needs hardware support (TLBs) to achieve acceptable average access times.

33. Why segmentation and paging may be combined into one scheme?

Segmentation and paging are often combined to improve upon each other. Segmented paging is helpful when page table becomes very large. A large contiguous section of page table that is unused can be collapsed into a single segment table entry with a page-table address of zero. Paged segmentation handles the case of having very long segments that require a lot of time for allocation. By paging the segments, we reduce wasted memory due to external fragmentation as well as simplify the allocation.

34. Why page size should be a power of 2?

The selection of a power of 2 as page size makes the translation of a logical address into a page number and page offset particularly easy. The page number and the offset is represented by several bits of the address. Splitting an address between bits results in a page size that is a power of 2.

35. What is contained in the page table?

Base address of each frame and corresponding page number.

36. How are the page numbers and offset numbers obtained?

Logical address is split into two parts: right-hand bits give the offset numbers, and left-hand bits give the page number.

37. Describe the page-to-frame translation.

Logical address is split into page offset and page number. Scan page table for page number, corresponding entry is the frame number. It is combined with page offset to give physical address.

38. How many frames are needed for each page?

One.

39. How much fragmentation occurs with paging? Which type?

On the average, one-half of last page in each job. It is internal fragmentation.

40. In what order are the frames assigned?

In the order to the free frames list.

41. List three ways of implementing the page table.

a. Dedicated registers.

b. In main memory.

c. Associative registers (cache).

42. What is an associative register?

It is a register in which no search is needed to find a key. All parts are given a key at once. The part holding the key is the only part to respond.

43. What is reentrant code?

It is a code that can be used by several users simultaneously. It cannot be altered during execution.

44. What is a TLB? What is its function?

A translation look-aside buffer is an associative cache of page table entries used to speed up the translation of virtual addresses to physical addresses.

45. Describe the algorithms for allocating regions of contiguous memory, and comment on their properties.

First-Fit

Scan memory region list from start for first fit.

Must always skip over potentially many regions at the start of list.

Best-Fit

Pick the closest free region in the entire list.

Tends to leave small unusable regions, and slower due to requirement of search the entire list.

Worst-Fit

Find the worst fit in the entire list

Slower as it searches the entire list, fragmentation is still an issue.

46. Explain the difference between internal & external fragmentation.

Internal fragmentation is the area occupied by a process but cannot be used by the process. This space is unusable by the system until the process releases the space.

External fragmentation exists when total free memory is enough for the new process but it's not contiguous and can't satisfy the request. Storage is fragmented into small holes.

47. What is the effect of allowing two entries in a page table to point to the same page frame in memory?

Allowing two entries in a page table to point to the same page frame in memory will decrease the time it would take to copy a large amount of data. We need to allow the new entries in page table point to the frames related to the data to be copied. But the effect is that when one part of the program modifies the content of shared frame, all pages related to it will be affected. Sometimes such effect is not desirable.

48. What is the mechanism in which one segment may belong to the address space of two different processes?

Segment tables are a collection of base-limit registers. Segments can be shared when entries in the segment table of two different jobs point to the same physical location. The two segment tables must have identical base pointers. The shared segment number must be the same in the two processes.

8

VIRTUAL MEMORY

Chapter Overview

- 8.1 Virtual Memory
- 8.2 Demand Paging
- 8.3 Page Faults / Segment Faults
- 8.4 Page Replacement
- 8.5 Page Replacement Algorithm
  - 8.5.1 First-In First-Out (FIFO)
  - 8.5.2 Page Optimal Replacement
  - 8.5.3 Least Recently Used (LRU)
  - 8.5.4 Not Recently Used (NRU)
  - 8.5.5 LRU Approximation Algorithms
    - 8.5.5.1 Additional-Reference Bits Algorithm
    - 8.5.5.2 Second-Chance Algorithm
    - 8.5.5.3 Enhanced Second-Chance Algorithm
- 8.6 Allocation of Frames
  - 8.6.1 Minimum Number of Frames
  - 8.6.2 Algorithms
  - 8.6.3 Replacement Scope
- 8.7 Thrashing
- 8.8 Working Set
- 8.9 Page Size

Review Questions

3:00

8.1 Virtual Memory

Virtual memory is a technique that is used to execute processes that are not in main memory completely. It allows to execute larger processes than main memory. The operating system places the currently required parts of a process in main memory. The remaining parts are placed on the backing storage. These parts are swapped into main memory as and when required. The backing store is usually an area of the disk outside of the file system.

If the flow of execution moves to a part that is not in memory, the operating system loads the required part from secondary storage into the memory. It allows to execute more processes simultaneously as the processes can be non-contiguous in paging system. Each process can be larger than the available real memory.

8.2 Demand Paging

Virtual memory can be implemented by a technique called **demanding paging.** It is a technique in which a Page is brought into memory when it is actually needed.

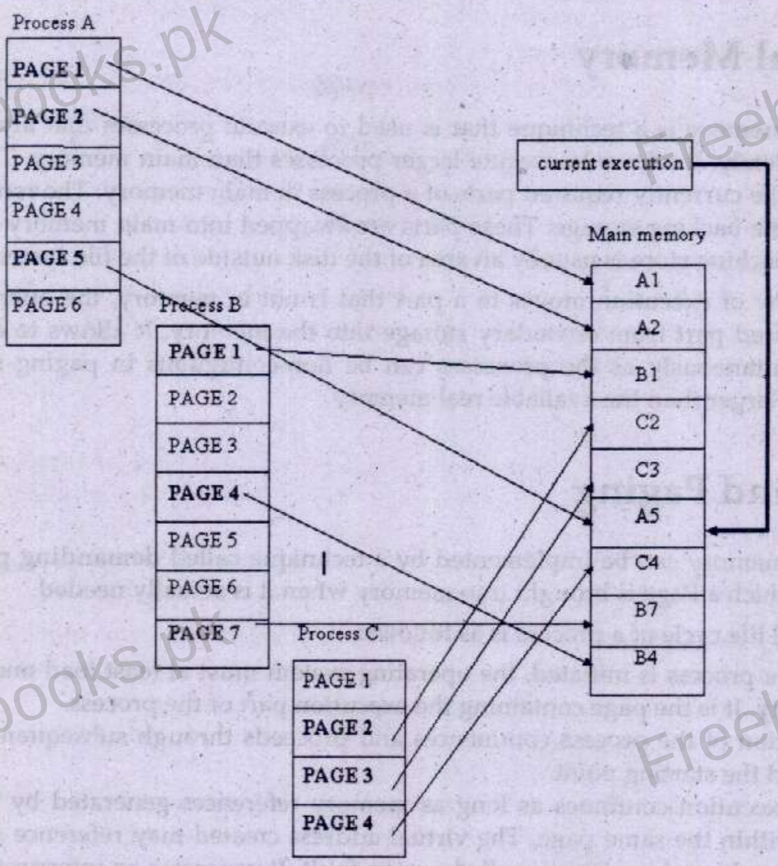
A typical life cycle of a process is as follows:

1. When a process is initiated, the operating system must at least load one page in real memory. It is the page containing the execution part of the process.
2. Execution of the process commences and proceeds through subsequent instructions beyond the starting point.
3. This execution continues as long as memory references generated by this page are also within the same page. The virtual address created may reference a page that is not in real memory. This is called a **page fault.** It generates an interrupt that asks for the referenced page to be loaded. This is called **demanding page.**
4. The operating system will try to load the referenced page into a free real memory frame. When this is achieved the execution can continue.
5. Finally when the process terminates, the operating system releases all the pages belonging to the process. The pages become available to other processes.

The next diagram shows virtual to real allocation for three processes A, B and C. Only some parts of the processes are loaded in real memory. Consider what happens if there is no free frame where to load a referenced page.

Suppose that the execution is at page A5 in the next diagram. If a new page A6 from process A is referenced then it must be loaded in real memory. But the memory is already full with pages.

In general, the operating system accommodates the new page by removing a currently loaded page that is not in use. This is called **page replacement.** It is important to remove a page that will not be accessed in a short time. It will reduce the number of page faults in the system.


Process A
PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
Process B
PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
Process C
PAGE 1
PAGE 2
PAGE 3
PAGE 4
current execution
Main memory
A1
A2
B1
C2
C3
A5
C4
B7
B4

8.3 Page Faults / Segment Faults

Page faults occur when a process references an address on a page that is not in memory. If there are no free frames in memory (which is usually the case), then a page of some active process from the memory must be evicted so that the page that was faulted for can be brought in.

It is important that the operating system chooses a good **page replacement policy**. If a page is evicted just before it will be needed again, it will also cause a page fault. If proper measures are not taken, it is possible for the system to experience **thrashing**.

8.4 Page Replacement

In a multiprogramming environment, almost all programs use probably half of their size. For example, a program of 10 pages actually uses only 5 pages. The demand paging mechanism saves the I/O time necessary to load the five pages that are never used. In this case, the degree of multiprogramming should be increased.

Increasing the degree of multiprogramming may lead to the situation where over allocation memory could occur.

If a page fault occurs, the operating system checks the internal page table. It determines where the desired page is on the disk but finds no more free frames for these required pages to be swapped in. The operating system has some options to solve this problem:

- Either it terminates the user program immediately; or

- It swaps out one program and frees all of its frames. It reduces the degree of multiprogramming.

Page replacement is another possibility in this situation. It frees one frame that is not currently used. It writes the contents of the frame to the disk and changes the page table to indicate this page is no longer in memory. The freed frame can now be used to hold the page for which the program faulted. The page fault processing procedure can be modified to include the page replacement mechanism:

- Find the location of the desired page on the disk
- Find a free frame. If there is a free frame, use it. Otherwise use a page-replacement algorithm to select a frame, write the selected page to the disk and change the page and frame tables accordingly.
- Read the desired page into the newly free frame; change the page and frame tables;
- Restart the user process.

If there is no free frame and the page replacement mechanism takes place, two page transfers are carried out. This doubles the page fault processing time and therefore increases the effective access time.

In order to reduce this overhead, the **modify bit** can be used. Each page or frame has a modify bit. Setting this bit to 1 indicates that the associated page has been modified since it was read from the disk. When we select a page for replacement, we check its modified bit first. If this bit is set, we must write this page to the disk before swapping in the desired page. Otherwise, we can avoid writing this page to the disk.

8.5 Page Replacement Algorithm

The algorithms used to select the page to be replaced are called **page replacement policies**.

8.5.1 First-In First-Out (FIFO)

The **First-in First-out** policy removes the page that has been resident in memory for the longest time. It assumes that such a page is no longer in use. It is also very simple to implement. In spite of being frequently required, the FIFO method will periodically remove pages. It causes another early page fault.

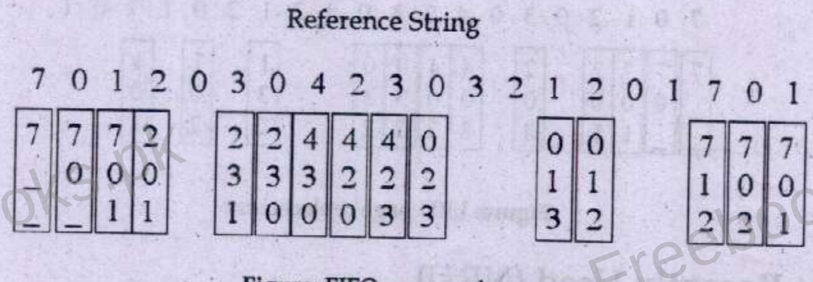


Figure: FIFO page replacement

In the above example, we have three empty frames and first three references (7,0,1) are brought in these frames. Now the next reference (2) replaces page 7 because it came first. The next reference is 0 that is already in the memory so no page fault occurs. This process continues as shown in the figure. Whenever a page fault occurs, we replace the page that entered the memory first of all.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

8.5.2 Page Optimal Replacement

This algorithm produces lowest page fault rate. It replaces the page that will not be used for the longer period of time.

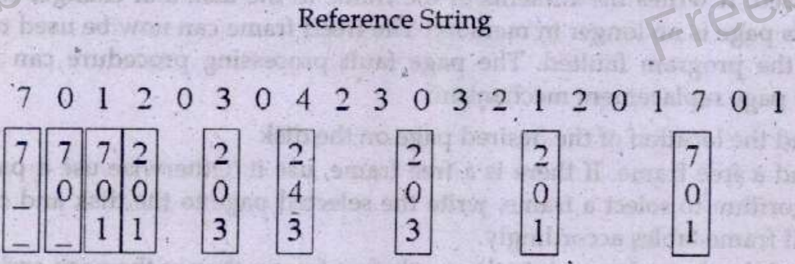


Figure: Optimal page replacement

If we apply this algorithm on the same previous example, it will produce only nine page faults. First three references produces page fault and fill three empty frames. The reference to page 2 replaces page 7 because it will not be used until reference 18. This algorithm is difficult to implement because we require future knowledge of all the references.

8.5.3 Least Recently Used (LRU)

Least Recently Used policy replaces the page whose time since last reference is greater. This requires a time stamp (marker) recording for a page frame at the time of each reference. The selected page for replaced would then have the oldest time. The overhead in such a system is that of maintaining all the time stamps (generally a linked-list) and the time taken to find the oldest value.

In the figure below, LRU Algorithm produces twelve faults. The first five faults are similar to optimal replacement. However, when the reference to page 4 occurs, LRU decides that page 2 was used recently. So it replaces page 2, although it is just going to be referenced again. Now page 2 is referenced and brought into memory by replacing page 3.

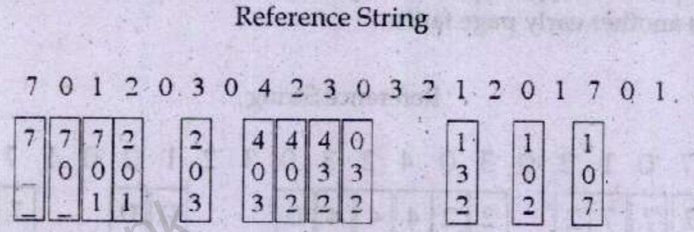


Figure: LRU page replacement

8.5.4 Not Recently Used (NRU)

Not Recently Used algorithm associates a 'page reference bit' with each page frame. This policy selects an interval of time during which the page reference bit is set to 1 if the page is referenced during this time. After the interval terminates, all the bits are reset and a new interval of time is set. Thus if a page has its page reference bit set to 1, it indicates that this page has been used during the current interval. The NRU policy simply selects for replacement any page with a page-referenced bit of 0.

8.5.5 LRU Approximation Algorithms

There are three LRU approximation page replacement algorithms. All make use of a reference bit associated with each entry in the page table. Initially the reference bit is set to 0. When the page is accessed (either read or write), the reference bit is set to 1. After some time the bits are checked and from this we know which pages have been used since the last check.

8.5.5.1 Additional-Reference Bits Algorithm

In this algorithm an 8-bit byte for each page is kept in a table in memory. At regular time intervals, the reference bits for each page are shifted into the high-order bit of its 8-bit byte, shifting the other bits right 1 bit and discarding the low-order bit. Then the reference bit is cleared. These 8-bit shift registers contain the history for the page use for the last eight time periods. If a page was recently used, the page will have a register like 10000000. If a page has not been used at all, it will have a register like 00000000. If the registers are interpreted as an unsigned integer, the page with the lowest number is the LRU page. If more than one page has the same LRU value then we can either swap all pages with that value or use a FIFO selection.

8.5.5.2 Second-Chance Algorithm

This algorithm is a FIFO replacement algorithm with consideration given to the reference bit. When a page has been selected for replacement, its reference bit is inspected. If it is set to 0, then it hasn't been used recently and it is replaced. If the reference bit is set to 1, then it is given a second chance and move on to the next FIFO page. When a page is given a second chance, its reference bit is cleared and its arrival time reset to the current time. This can easily be implemented using a circular queue. If all reference bits are set to 1 then the algorithm degenerates to FIFO.

8.5.5.3 Enhanced Second-Chance Algorithm

In this algorithm, the method is the same as for second chance but the reference bit and the modify bit are used as an ordered pair. This gives 4 classes of pages:

1. (0,0) - Neither recently used nor modified - best replacement choice.
2. (0,1) - Not recently used by modified - not quite as good as the page will need to be written out before replacement.
3. (1,0) - Recently used but clean - it will probably be used again soon.
4. (1,1) - Recently used and modified - it will probably be used again soon and it needs to be written out before replacement.

When a page replacement is called for, a page will belong to one of these four classes. The page selected is the first one encountered in the lowest non-empty class. So when we are looking for a page to replace, we may need to cycle through the circular array several times before we find a page to be replaced. First time through, if encounter a page in a class (1,x) then give a second chance and reset the reference bit. When a (0,x) class page is found (i) and if it is (0,0) then replace. If it is (0,1) then may mark it and look further to see if there is a (0,0).

8.6 Allocation of Frames

Normally, there are fixed amounts of free memory with various processes at different time in a system. The question is how this fixed amount of free memory is allocated among the different processes.

The simplest case is the single process system. All available memory for user programs can initially be put on the free frame list (pure demand paging). When the user program starts its execution, it will generate a sequence of page faults. The user program would get all free frames from the free frame list. As soon as this list was exhausted and the more free frames are required, the page replacement algorithm can be used to select one of the in-used pages to be replaced with the next required page and so on. After the program was terminated, all used pages are put on the free frame list again.

The frame allocation procedure is more complicated when there are two or more programs in memory at the same time.

8.6.1 Minimum Number of Frames

Of course, we cannot allocate more than the total number of available frames in the system. On the other hand, there is a minimum number of frames which must be allocated. This minimum number is determined by the instruction architecture. It is obvious that we must provide enough frames to hold all the different pages that any single instruction can reference. For example, all memory reference instructions of a machine have only one memory address. So we need at least one frame for the instruction code and one frame for the memory reference. If one level indirect addressing is allowed, a load instruction on page *m* can refer to an address on page *j*. It is an indirect reference to page *k*. We need three pages.

8.6.2 Algorithms

The simplest way is to divide *m* available frames among *n* processes to give everyone an equal share, *m/n* frames. This is called **equal allocation**. Various processes will need different amounts of memory. If the equal allocation is applied, there can be some frames wasted. Therefore, other allocation scheme can be used to give available memory to each process according to its size. This is called **proportional allocation**. Let the size of the virtual memory for process p_i be s_i, the number of frames allocated to the process p_i be a_i, and define

$$S = \Sigma s_i$$

If the total number of available frames is *m*, then a_i can be calculated:

$$a_i = (s_i/S) * m.$$

Of course a_i must be adjust to be a integer, greater than the minimum number of frames required by the instruction set with a sum not exceeding *m*.

In both of theses cases, the number of frames allocated to each process may vary according to the multiprogramming level, say *l*. If *l* increases, each process will lose some of the allocated frames to provide memory needed for the new process. Otherwise, the frames allocated to the departed process can be now spread over the remaining processes.

Within these two allocation schemes, a high-priority process is treated the same as low-priority process. By definition, it is desirable to give more memory to high-priority process to speed up its execution.

8.6.3 Replacement Scope

When its necessary to find free page frames, what set of pages should become candidates for replacement?

- Local replacement policies replace pages that belong to the process that needs the new frame.
- Global policies consider all unlocked frames. Most systems use global replacement because it is easy to implement, has minimal overhead, and performs reasonably well.

	Local Replacement	Global Replacement
Fixed Allocation	Rarely used –A process is given a fixed number of frames. Page faults are satisfied from this set.	This combination isn't possible
Variable Allocation	The process is given a fixed allocation and pages to be replaced are chosen from this set. Periodically, the resident set size is re-evaluated. Pages can be added or subtracted.	Replacement pages are chosen from any page in memory. Resident set size varies, although by a blind process. This is the most common approach

8.7 Thrashing

The virtual memory model gives the opportunity for the system to sustain an indefinite number of processes simultaneously. If too many processes are running, their resident-in-real-memory pages will be very restrictive. It will cause frequent occurrence of page faults. At some time, the processor may be spending the most of its time swapping pages and doing little productive work. This condition is known as **thrashing.**

Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.

The below graph shows CPU utilization rate with the number of active processes:

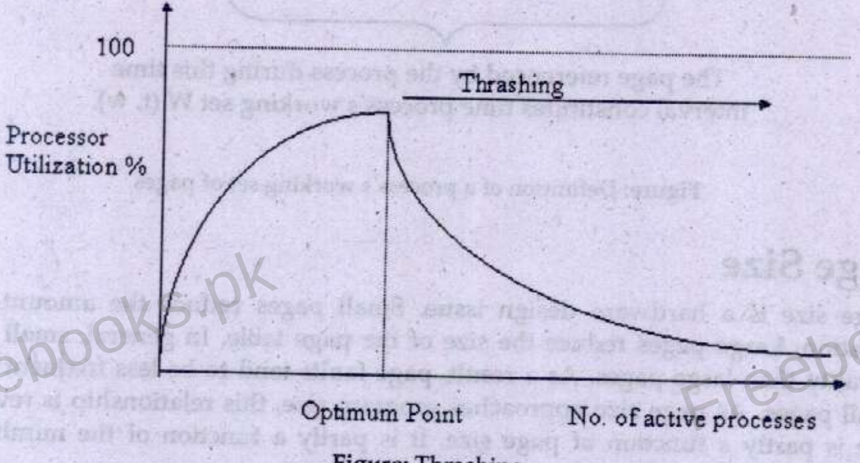


Figure: Thrashing

From the above we see that the operating system must regulate the number of active processes to prevent the onset of thrashing.

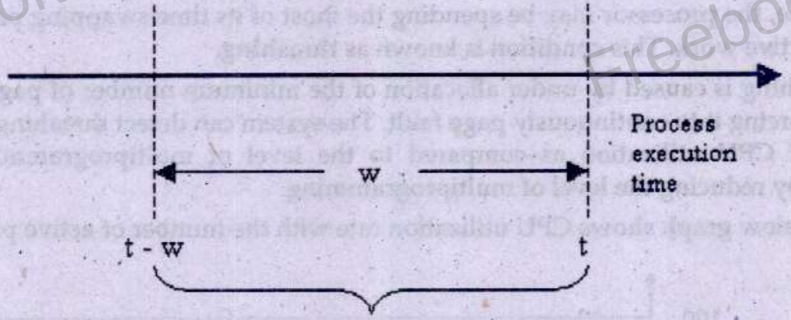
8.8 Working Set

A working set is a collection of pages, which is actively referenced by a process. The working set of processes should be kept in main memory to run efficiently. Otherwise thrashing may occur repeatedly. A popular rule to avoid thrashing is to give processes enough page frame to hold half of their virtual space.

A working set storage management policy is used to maintain the working sets of active program in main memory. The decision to add a new process to the active set of processes is decided on the basis of whether enough space is available in the main memory to accommodate the working set of pages for the new process.

The working set of a process W (t,w) at time t is the set of pages referenced by process during the process time interval t. process time means the time during which a process has the CPU. The variable w is called the working set window size. The working set storage management policy uses this size for effective operation.

Working sets change during process execution. Pages are added or deleted and critical changes may occur as the process requires a completely different working set. The initial and subsequent working sets of a process may differ completely in size and contents. It complicates the storage management under a working set strategy.



The page referenced by the process during this time interval constitutes time process's working set W (t, w).

Figure: Definition of a process's working set of pages

8.9 Page Size

Page size is a hardware design issue. Small pages reduce the amount of internal fragmentation. Large pages reduce the size of the page table. In general, small pages have better locality than large pages. As a result, page faults tend to be less frequent for systems with small pages. As page size approaches program size, this relationship is reversed. Page fault rate is partly a function of page size. It is partly a function of the number of pages allocated to a process.

Review Questions

1. What is virtual memory?

Virtual memory is a technique to execute processes that may not be completely in main memory. It abstracts main memory into an extremely large storage. This array of storage is mapped on to main memory and its backing store. The backing store is usually an area of the disk outside of the file-system.

2. What causes page fault? What actions may be taken by O/S when servicing page fault?

A page fault occurs when an access to a page that has not been brought into main memory takes place. The operating system verifies the memory access. It aborts the program if it is invalid. If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted.

3. Indicate the states of instruction execution when a page fault occurs.

a. Page fault while fetching the instruction.

b. Page fault while fetching the operands.

c. Page fault while storing data to memory.

4. List cases where entire program need not be in memory, traditionally.

a. Certain options of a program that are rarely used.

b. Many error-handling sections.

c. Large arrays, lists, and tables, where only a small portion is used.

5. List benefits of having only part of a program in memory.

a. Simplifies the programming task.

b. More user-programs can run concurrently in newly freed memory.

c. Less swapping of entire programs, thus less I/O.

6. What is meant by demand paging and pure demand paging?

Demand paging is a mechanism to utilize virtual memory. When a process is to be executed, a procedure called **lazy swapping** loads in only part of the process so that it can start running. Other pages are brought into main memory when needed. In pure demand paging, a process starts execution with no pages in memory and page loading is triggered by the page fault trap as they are required.

7. List the advantages of demand paging.

It decreases swap time and the amount of free physical memory.

It allows higher degree of multiprogramming.

8. What is the dirty bit and what is it used for? Why is it used?

The dirty bit is an extra bit associated with a page that is set when a page has been modified. It is used to make page replacements quicker. If a page has not been modified then there is no need to write it back to the backing store. The page can simply be marked as free and then new page loaded. This reduces the page replacement operation for clean pages from 2 disk access to 1.

9. Consider the following page reference string:

3,2,5,1,2,3,4,6,1,7,2,1,6,1,3,7,3,2,1,7

How many page faults would occur for LRU, FIFO and Optimal replacement algorithms, assuming three, four, five or six frames? Using pure demand paging.

Frames	LRU	FIFO	Optimal
3	16	16	10
4	12	10	8

5	9	10	7
6	7	9	7

10. List the costs and benefits of implementing virtual memory

The costs are additional hardware and slower access time. The benefits are good utilization of memory and larger logical address space than physical address space.

11. What factors determine the page-fault time?

a. Time to service interrupts.

b. Time to swap page.

c. Time to restart process.

12. List ways of resolving problem of no free frames left.

a. Aborting user program (poor solution).

b. Swap out an entire program, freeing its frames.

c. Replacing particular existing frames.

13. What is page replacement?

It is a process that selects a frame (preferably not in use) as a victim. It swaps it out and swap in the desired page into this frame. It then restarts the program.

14. How many swaps are needed for pure page replacement?

Two: first one out, second one in.

15. Ideally, what criteria should we use to replace pages?

Choose the victims to achieve the lowest page-fault rate.

16. What is Belady's anomaly?

You may expect that the page-fault rate would decrease as the number of frames increases. But Belady's anomaly says this is not true in all algorithms.

17. What is the ideal page-replacement scheme?

Optimal or minmum page-fault method. Replace the page that will not be used for the longest future time.

18. Define two-segment replacement algorithm based on FIFO page-replacement scheme.

Find the first segment large enough to accommodate the incoming segment. If relocation is not possible and no one segment is large enough, select a combination of segments whose memories are contiguous, which are "closest to the first of the list" and which can accommodate the new segment. If relocation is possible, rearrange the memory so that the first *N* segments large enough for the incoming segment are contiguous in memory. Add any leftover space to the free-space list in both cases.

19. Define two-segment replacement algorithms based on LRU page-replacement scheme.

It selects the segment that has not been used for the longest period of time and that is large enough. It adds any leftover space to the free space list. If no one segment is large enough, it selects a combination of the "oldest" segments that are contiguous in memory (if relocation is not available) and that are large enough. If relocation is available, it rearranges the oldest *N* segments to be contiguous in memory and replace those with the new segment.

20. What is the cause of thrashing?

Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.

21. What is the difference between local and global page allocation? What are their respective advantages and disadvantages?

When a process needs a page frame, a local allocation scheme will always take the page frame away from another page of the same process. A global scheme can take it away from any process.

Global allocation systems respond quickly to changes in working set sizes. The local allocation systems cannot respond at all.

Programs that try to optimize their occupancy of memory have a more predictable environment in a local allocation system. Under global allocation system, they cannot control the amount of memory allocated to them.

22. What effect does increasing the page size have?

- Increases internal fragmentation
- Decreases number of pages
- Increases TLB coverage
- Increases page fault latency
- Increases swapping I/O throughput

23. Describe four replacement policies and compare them.

Optimal

- Toss the page that won't be used for the longest time
- Impossible to implement
- Only good as a theoretic reference point: The closer a practical algorithm gets to optimal, the better

FIFO

- First-in, first-out: Toss the oldest page
- Easy to implement
- Age of a page is isn't necessarily related to usage

LRU

- Toss the least recently used page
- Assumes that page that has not been referenced for a long time is unlikely to be referenced in the near future
- Will work if locality holds
- Implementation requires a time stamp to be kept for each page, updated on every reference
- Impossible to implement efficiently

Clock

- Employs a usage or reference bit in the frame table.
- Set to one when the page is used
- While scanning for a victim, reset all the reference bits
- Toss the first page with a zero reference bit.

9

FILE SYSTEM MANAGEMENT

Chapter Overview

9.1 File
    9.1.1 File Attributes
    9.1.2 File Operations
9.2 Open Files
9.3 Types of File System Objects
9.4 Information Types
9.5 File Management Systems
9.6 File System Architecture
    9.6.1 File Management Functions
9.7 Access Methods
    9.7.1 Sequential Access
    9.7.2 Direct Access
    9.7.3 Other Access Methods
9.8 File Locking
9.9 File Blocking
9.10 File Protection
    9.10.1 File Naming
    9.10.2 Passwords
    9.10.3 Access Control
9.11 File system Reliability
    9.11.1 Periodic Dump
    9.11.2 Incremental Dump
9.12 File Management Module
    9.12.1 Accessing and Organizing
    9.12.2 Management of Secondary Memory Space
        9.12.2.1 Contiguous Blocks
        9.12.2.2 Block Linkage
        9.12.2.3 Index Blocks
9.13 Directories and Names
9.14 Pre-Process Root Directory
9.15 Directory Structure
    9.15.1 Operations Performed on Directory
    9.15.2 Single Level Directory
    9.15.3 Two-Level Directory
    9.15.4 Tree-Structured Directory
    9.15.5 Acyclic-Graph Directories
    9.15.6 General Graph Directories
9.16 Directory Entries

Review Questions

9.1 File

The information can be stored on different storage media like optical disks, magnetic disks and magnetic tapes etc. The operating system provides a uniform logical view of the information storage. The logical storage unit defined by the operating system is called file.

A **file** is a named collection of related information stored on secondary storage. A file is the smallest allocation of logical secondary storage from user point of view. The data can be stored on secondary storage only if it is in a file. The files commonly represent programs and data. The data files can be numeric, alphabetic, alphanumeric or binary. A file is generally a sequence of bits, bytes, lines or records.

The information stored in a file is defined by its creator. A file may contain different types of information such as source program, object program, numeric data or text etc. A file has a specific structure that depends on its type. For example, a **text file** is a sequence of characters organized into lines and a **source file** is a sequence of subroutines and statements. Similarly, an **object file** is a sequence of bytes organized into blocks that can be understood by the linker.

9.1.1 File Attributes

Any file system needs to track the following minimum attributes of files:

File Name	It is the file name in human-readable form of file identification.
File Identifier	It is usually a number that is not for human use. It identifies the file within the file system.
Type	It describes the type of information stored in file.
Location	It is a pointer to file location on a device.
Size	It indicates the current size of the file in bytes, words or blocks. It may also include the maximum allowed size for the file.
Protection	It describes the users who can access the file. The type of access can be reading, writing, executing etc.
Time, date, and user ID	The time describes the time of creation, last access and last update time. It can be used for protection, security and usage monitoring.

9.1.2 File Operations

A file can be considered as an abstract data type. The following abstract operations can be performed on files:

Creating File	It requires two steps. First, space must be found for the file in the file system. An entry for the new file must be made in the directory.
Writing File	A system call is used to write a file. A file identifier is provided with the information to be written. The operating system performs the write operation. The operating system must maintain the write pointer in the file.
Reading File	A system call is used to read a file. A file identifier is provided with the address to read data. The operating system handles the read operation. The operating system must maintain the read pointer in the file.

Repositioning in File	It is a process of moving the read/write pointer in the file or moving through the directory entries.
Deleting File	The operating system must provide the facility to delete a file. It frees the space allocated to the file and removes the file entry from the directory.
Truncating File	It means that the file contents are erased but the directory entry is kept intact. The file will effectively be empty.

9.2 Open Files

The operating system usually maintains several lists for open files. There is a global list that contains references to open files all across the system for all processes. There is also a local list that maintains open files per process.

Depending on the setup, these lists and data structures will contain the following:

File Pointer	The read/write pointer inside the file.
File Open Count	In the global table, the system needs to maintain how many instances of this file are currently opened by various processes.
Disk location of the file	The hard drive location of the file is usually kept in memory. It avoids the reading of the hard drive to save something to a file.
Access Rights	Each file is opened in some access mode such as read only etc. An operating system can store this in a per-process open file table so that it can deny or allow some file operation on a per-process basis.

9.3 Types of File System Objects

Different types of object in a file system are as follows:

Shortcut or Soft Link	Shortcut is a pointer to another's name in the file system.
Device	It is a hardware device like parallel port etc.
Pipe	Pipe is a communication channel between two processes. One process sends data to pipe and other process reads that data from pipe. The buffer used as pipe is limited in size. If it is full, a process cannot send data to it. Pipes can be named or unnamed. The unnamed pipes can only be used by the processes that create them.
Shared Memory	It is an allocation of memory locations for use by one or more processes.
Semaphore	A semaphore object that can be used to control synchronization of processes.

9.4 Information Types

Different files store different types of data. File types refer to the type of information stored in the file. File can be executable program, object file, C source code, word processing file and graphic etc. Files can be identified in three ways:

1. Name Extension

The name extension scheme identifies the information type by including a particular extension as part of the filename. for example, '.exe' and '.com' mean executable program and '.doc' means word document.

2. Magic Number

Magic number scheme identifies information type by looking for a magic number. This number may be a byte or sequences of bytes at a particular location in the file. For example, the string "GIF8" at the start of a file means that the file contains a GIF image data. Normally operating systems do not support this mechanism. A particular application may use it.

3. Operating System Supported Information Type

In this mechanism, operating system maintains the type as part of the information associated with file. The type of the file is set when it is created.

9.5 File Management Systems

A **file management system** is a set of system software that provides services to users and applications in the use of files. It allows the users and applications to access files.

Different objectives of a file management system are as follows:

- Fulfilling the data management needs and requirements of the user including storage of data and the ability to perform different operations
- Ensuring that the data in the file are valid
- Optimizing performance from system point of view in terms of overall throughput and from the user's point of view in terms of response time
- Providing I/O support for a variety of storage device types
- Minimizing or eliminating the potential for lost or destroyed data
- Providing a standardized set of I/O interface routines to user processes
- Providing I/O support for multiple users in case of multiple-user systems

The minimum requirements for interactive and general-purpose system are as follows:

1. Each user should be able to create, delete, read, write and modify files.
2. Each user may have controlled access to other users' files.
3. Each user may control the types of accesses allowed to the user's files.
4. Each user should be able to restructure the user's files in a form that is appropriate to the problem.
5. Each user should be able to move data between files.
6. Each user should be able to back up and recover the user's files in case of damage.
7. Each user should be able to access his or her files by name rather than by numeric identifier.

9.6 File System Architecture

The **device drivers** communicate directly with peripheral devices or their controllers or channels at the lowest level. A device driver is responsible for starting I/O operations on a device and processing the completion of I/O request. The typical devices that are controlled for file operations are disk and tape drives. The device drivers are usually often considered part of operating system.

The next level is known as the **basic file system** or **physical I/O level**. It is the basic interface for the environment outside the computer system. It deals with the blocks of data that are exchanged with disk or tape systems. It is related to the placement of those blocks on secondary storage device and on the buffering of these blocks in main memory. However, it does not understand the contents of data and the file structure. The basic file system is often considered a part of operating system.

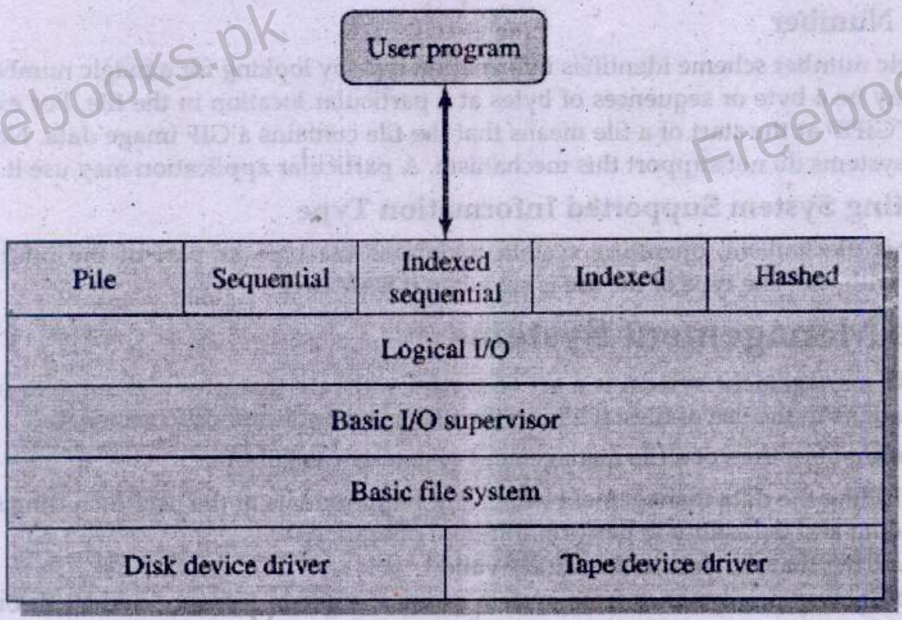


Figure: File System Software Architecture

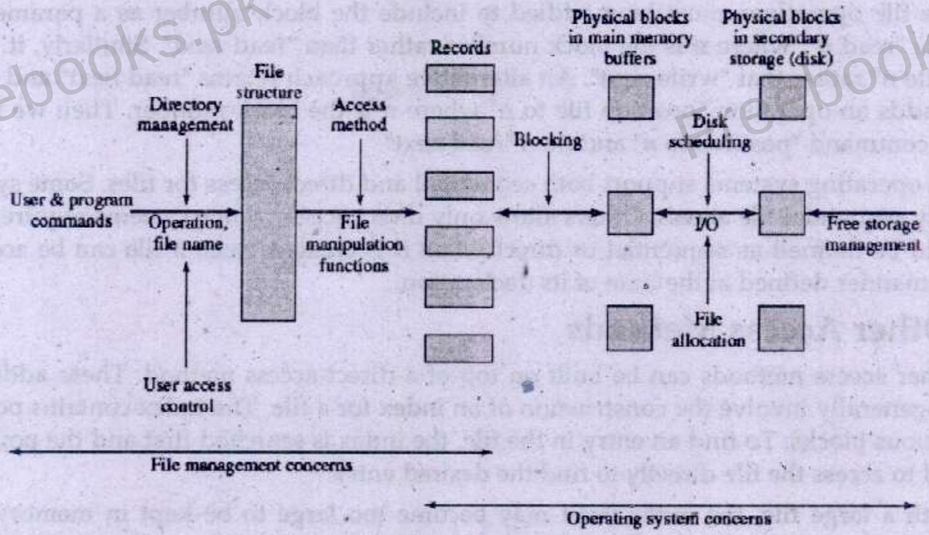
The **basic I/O supervisor** manages all file I/O initialization and termination. The control structures at this level are maintained for device I/O, scheduling and file status. The basic I/O supervisor selects the device on which file I/O is to be performed depending on the selected file. It is also related to scheduling disk and tape accesses to optimize performance. I/O buffers are assigned and secondary memory is allocated at this level. The basic I/O supervisor is a part of the operating system.

The **logical I/O** enables users and applications to access records. The logical I/O module deals with file records. It provides general-purpose record I/O capability and also maintains the basic data about the files.

The **access method** is the file system's closest level to the user. It provides a standard interface between applications, file systems and devices that hold the data. Different access methods reflect different file structures and different ways to access and process data.

9.6.1 File Management Functions

The users and application programs interact with the file system using different commands to create and delete files and perform operations on files. The file system must identify and locate selected file before performing any operation. It requires a directory that is used to describe the location of all files and their attributes. Most shared systems also apply user access control and only authorized users can access particular files. The user or application may perform the basic operations on a file at record level. The user views the file as a structure that organizes the records such as a sequential structure. An access method is required to translate user commands into specific file manipulation commands.


Physical blocks in main memory buffers
Physical blocks in secondary storage (disk)
Records
File structure
Directory management
Access method
Blocking
Disk scheduling
User & program commands
Operation, file name
File manipulation functions
I/O
Free storage management
File allocation
User access control
File management concerns
Operating system concerns

However, I/O is performed on a block. The records or fields of a file must be organized as a sequence of blocks for output and unblocked after input. Many functions are required to support block I/O. The secondary storage must be managed and it involves allocating files to free blocks on secondary storage. It also requires to manage free storage to know the blocks that are available for new files. The individual block I/O requests must also be scheduled. Both disk scheduling and file allocation are related to optimize performance and should be considered together. The optimization will depend on file structure and the access patterns.

9.7 Access Methods

There are several ways to access information in a file. Some systems provide only one access method for files. Some systems provide different access methods. Selection of the right access method for a particular application is a major design problem.

9.7.1 Sequential Access

Information in the file is processed in order, one record after the other. This is by far the most common mode of access of files. For example, computer editors usually access files in this fashion.

A read operation reads the next portion of the file and automatically advances the file pointer. Similarly, a write operation appends to the end of the file and advances to the end of the newly written material i.e. the new end of file. Such a file can be reset to the beginning. On some systems, a program may be able to skip forward or backward n records for some integer n. This scheme is known as **sequential access**. Sequential access is based on a tape model of a file.

9.7.2 Direct Access

Direct access is based on a disk model of a file. For direct access, the file is viewed as a numbered sequence of block or records. A direct-access file allows arbitrary blocks to be read or written. After block 18 has been read, block 57 could be next and then block 3. There are no restrictions on the order of reading and writing for a direct access file. Direct access files are of great use for intermediate access to large amounts of information.

The file operations must be modified to include the block number as a parameter. It works like "read n", where n is the block number rather than "read next". Similarly, it writes with "write n" rather that "write next". An alternative approach retains "read next" and "write next". It adds an operation "position file to n" where n is the block number. Then we would issue the command "position to *n*" and then "read next".

All operating systems support both sequential and direct access for files. Some systems allow only sequential file access. Others allow only direct access. Some systems require that a file should be defined as sequential or direct when it is created. Such a file can be accessed only in a manner defined at the time of its declaration.

9.7.3 Other Access Methods

Other access methods can be built on top of a direct-access method. These additional methods generally involve the construction of an index for a file. The index contains pointers to the various blocks. To find an entry in the file, the index is searched first and the pointer is then used to access the file directly to find the desired entry.

With a large file, the index itself may become too large to be kept in memory. One solution is to create an index for the index file. The primary index file would contain pointers to secondary index files that would point to the actual data items.

For example, IBM's indexed sequential access method (ISAM) uses a small master index that points to disk blocks of a secondary index. The secondary index blocks point to the actual file blocks. The file is kept sorted on a defined key. We first make a binary search of the master index to find a particular item. It provides the block number of the secondary index. This block is read in. Binary search is used again to find the block containing the desired record. Finally, this block is searched sequentially. In this way, any record can be located from its key by at most direct access reads.

9.8 File Locking

File locking enables a process to implement mutually exclusive access to a file. File locking can be implemented in the following three ways:

- A file can be locked as a whole or partially.
- File locking may be applied on any access or different levels. Some systems provide both **read** and **write** locks. If a file is locked for reading, other processes can read it but cannot write. If a file is locked for writing, other processes cannot read or write it.
- Locks can be **mandatory** or **advisory**. In mandatory lock, operating system does not allow to access the file while it is locked. In advisory lock, locking primitives provide information about file's lock status.

9.9 File Blocking

Block is the unit on which I/O system performs I/O operations. Usually, the block size is a multiple of disk sector size. Blocking is a method that determines how records in a file are placed in blocks.

Types of Blocking

Different types of blocking are as follows:

Fixed Blocking	In fixed blocking, the blocks are of same size. Each block contains an integral number of records. If the record and block size is given, operating system can compute block and offset in block of a record.

Un-spanned Blocking	It is used when records are of variable size. Multiple records can be stored in each block but a record cannot be stored in multiple blocks. It means that record size should not be greater than block size. If a record is less than the block and the next record does not fit in the remaining space, it is wasted.
Spanned Blocking	In spanned blocking, records can be stored in multiple blocks. There is no restriction on block size and block space is not wasted. In order to compute a record's location, the size of all records has to sum that precede it.

9.10 File Protection

When designing file systems, an important consideration is how to protect sharable files. Sometimes this is avoided by not permitting sharing of files. So the protection is quite easy to implement or make all files public and provide complete sharing. However, it is not acceptable to the users. Users want to share some files. There are some files that should be accessible to only a group of users. The solution is to allow limited sharing. It means that users can share but to a limited extent only.

Protection is achieved by limiting the type of file access that can be made. Access is permitted or denied depending upon several factors. One is the type of access requested. Several operations on files can be controlled. Some of these are as follows:

Read	It reads information contained in the file
Write	It writes new information into the file at any point or overwrite existing information in the file.
Execute	It loads the contents of a file in main memory and create a process to execute it.
Append	It appends information at the end of a file.
Delete	It deletes a file and release its storage space for use in other files.
List	It lists the name and attributes of the file

The most common implementation of file systems allows the owners of the file to perform first five operations. Other users can only invoke those operations that do not modify the file such as file read. In some systems like UNIX, the user can change the access control of a file so that anybody can access or modify the file. The user can also completely deny any user access to a file including himself.

The three most popular implementations of file protection are the following:

9.10.1 File Naming

This depends upon the inability of a user to access a file he cannot name. This can be implemented by allowing only users to see the files they have created. Many file systems allow only a limited number of characters for filenames. There is no guarantee that two users will not use the same filenames.

9.10.2 Passwords

This scheme associates a password to each file. If a user does not know the password associated to a file then he cannot access it. This is a very effective way of protecting files for a user who owns many files and constantly changes the password. In this way, he makes sure that nobody accesses these files.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

9.10.3 Access Control

Access control limits who can access files and how they can access them. The simplest operating system access control mechanism gives unlimited access to all users. This is the access control scheme used by DOS. On such a system , users wishing to control access to their file must do so by limiting physical (and network) access to their machine

An access list is associated to each file or directory. The access list contains information on the type of users and accesses that they can do on a directory or file. An example is the following access list associated to a UNIX file or directory:

drwxrwxrwx

The **d** indicates that this is an access list for a directory. The first **rwx** indicates that it can be read, written and executed by the owner of the file. The second **rwx** is an access information for users belonging to the same group as the owner. The last **rwx** is for all other users. The **rwx** can be changed to just r-- indicating that it can only be read. It can be -w- for write-only and --x for execute only.

9.11 File system Reliability

On of the major problems in maintaining a file system is how to protect it for being corrupted by both systems failure and errors in system software's. The usual way of insuring reliability is by making duplicates (or even more) copies of files. This can be done by:

9.11.1 Periodic Dump

All files in the file system are copied to another device usually a magnetic tape. This is done regularly and may be done as often as weekly if the files are highly volatile.

9.11.2 Incremental Dump

All files that get modified from the last dump is written to another device. This is faster to do and usually done more often.

Although these methods are acceptable to most systems, it is possible to lose some data. This happens when the system crashes just before a dump was made. This means that all the modifications since the last dump will be lost. Hence, dumping the file system is not satisfactory in systems where no data loss can be tolerated. In this type of systems transaction logging is appropriate.

In transaction logging every transaction received by the system is dumped. If the system fails, the dumped transactions are run through the previously dumped files. This is usually used together with either periodic or incremental dump.

Another important aspect of a file system is its consistency. When a system crash occurs, it is possible that a modified block may not have been written yet. The standard way of checking for consistency is to read all the files and the blocks allocated to the files of the file system and then compare this with the blocks in the free list.

The action of the system is to allocate the missing block and append this to one of the files. The owner of the said file is then alerted of this addition to his file. Although it creates a problem to the file, the file system is consistent.

9.12 File Management Module

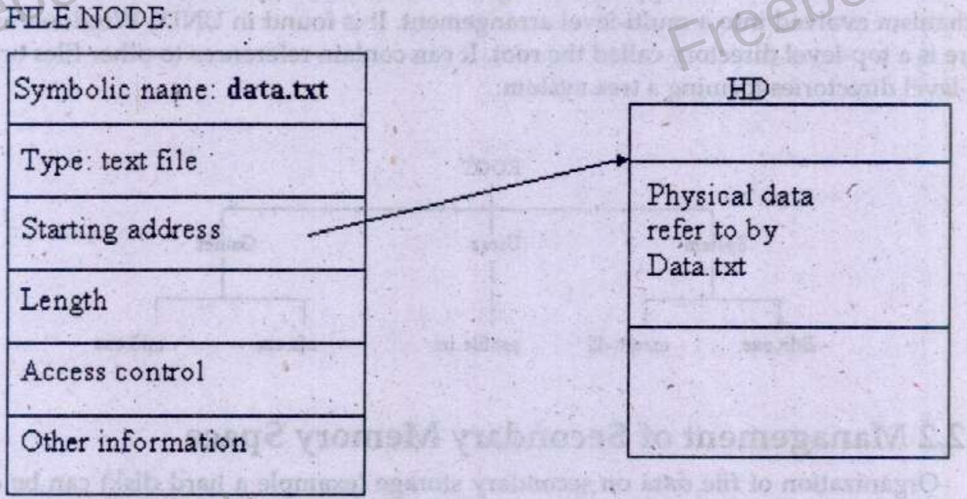
The majority of computer applications use storage facilities provided by magnetic disk and other storage devices to store non-volatile data. A *File* is the term used for a logical representation of this non-volatile data.

The **File Management Module** is the operating system component that manages all the files on secondary storage. The File Management Module provides a filing system for the following three functions:

1. It provides a means of organizing and accessing files (data) in a convenient way.
2. Perform automatic management of secondary memory space. The exact whereabouts of files in secondary storage should be of no concern to the user.
3. Protect the files against system failure or access by unauthorized users.

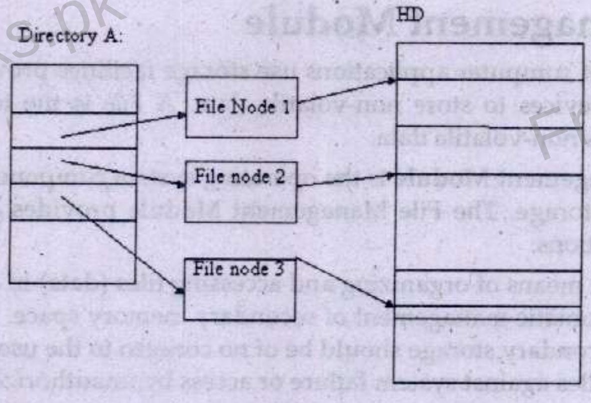
9.12.1 Accessing and Organizing

The basic problem of accessing a file for the OS is to hide the major complexities involved in storing the file data on secondary storage. Suppose a source program consists of lines of text terminated by new-line characters. The programmer using this file need not be concerned about how the file is held on disk and how it is packed into physical storage units on the disk surface. He must be able to access the file through a **symbolic name**.



The **symbolic name** of a file is the logical view by the user of the physical data on secondary storage. The operating system will keep track of the physical data by keeping a structure (called a **file node**) containing file information as shown above:

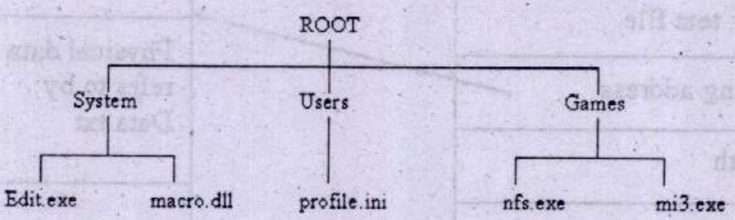
The file node will be kept in memory during the operating system run-time for fast access although it must be stored itself on secondary storage once the system goes down. Nowadays operating systems group file nodes in collections called **directories**. A **directory** is a logical grouping of files.


Directory A:
HD
File Node 1
File node 2
File node 3

Note:

1. Besides the group of files that it contains the directory structure can hold generic information on who can access the particular files.
2. Like the file nodes, the directory structure is itself stored on secondary storage once the system goes down.
3. The symbolic name of the file will consist of the directory name plus the file name for example "dir1\data.txt".

As the number of applications grew in modern systems, the above directory mechanism evolved into a multi-level arrangement. It is found in UNIX, Windows and DOS. There is a top-level directory called the root. It can contain references to other files or to other low-level directories forming a **tree system**:

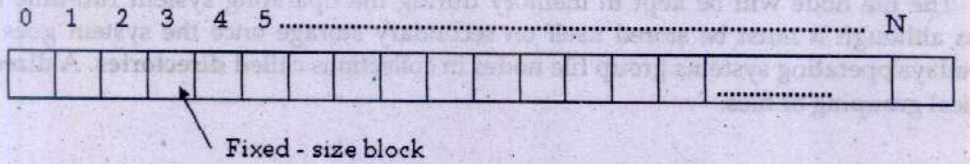

ROOT
System
Users
Games
Edit.exe
macro.dll
profile.ini
nfs.exe
mi3.exe

9.12.2 Management of Secondary Memory Space

Organization of file data on secondary storage (example a hard disk) can be different from the logical organization of files by the operating system. Secondary storage is typically organized into *fixed-sized blocks*, which are a multiple of 512 bytes.

A file will be made of one or more of these blocks:

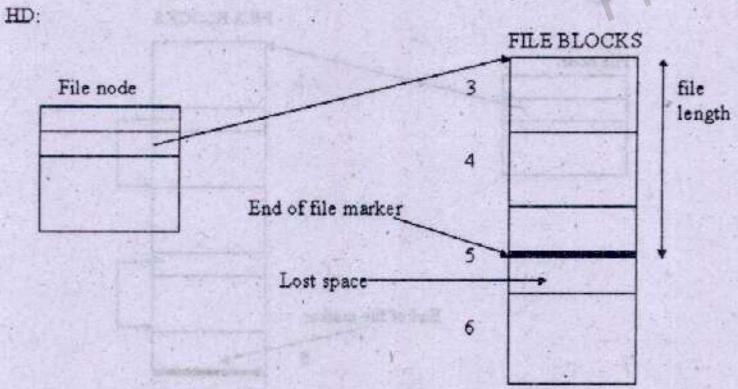
Hard Disk:


0 1 2 3 4 5 .... N
....
Fixed - size block

Three major techniques by which file data is organized in these blocks are as follows:

9.12.2.1 Contiguous Blocks

In this technique, the file data is stored in blocks adjacent to each other. The file node will point to the first block and will contain the length of the file. The whole of each block will be used to hold user data:



Advantages of Contiguous Blocks

The advantages of contiguous blocks are as follows:

- Simple to implement.
- Damage to a single block results in only localized loss of data.
- Any block may be accessed at random making contiguous files ideal for applications such as databases.

Disadvantages of Contiguous Blocks

The disadvantages of contiguous blocks are as follows:

- This technique suffers from fragmentation. As files are created and deleted, the free space becomes broken up into small pieces. No piece may be large enough by itself to hold anything but the smallest file.
- It is difficult to add and delete blocks in the middle of a file because it has to move all following blocks.

9.12.2.2 Block Linkage

In this technique a few bytes in each block are used as pointers to the next block. The last block will contain a null pointer (typically zero). The file node will point to the first block:

Advantages of Block Linkage

The advantage of block linkage is as follows:

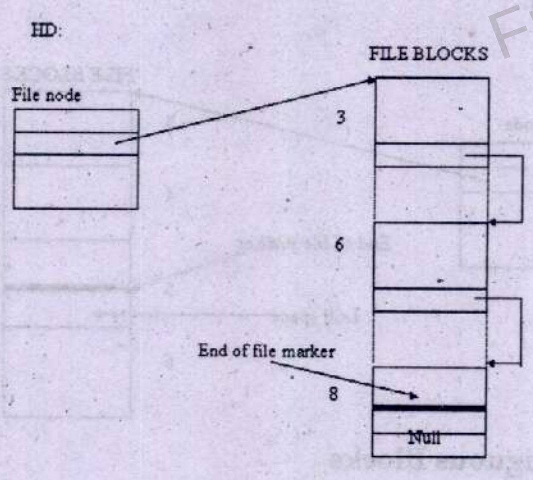
- Fragmentation is eliminated since now any block can be reserved for the file.

Disadvantages of Block Linkage

The advantages of block linkage are as follows:

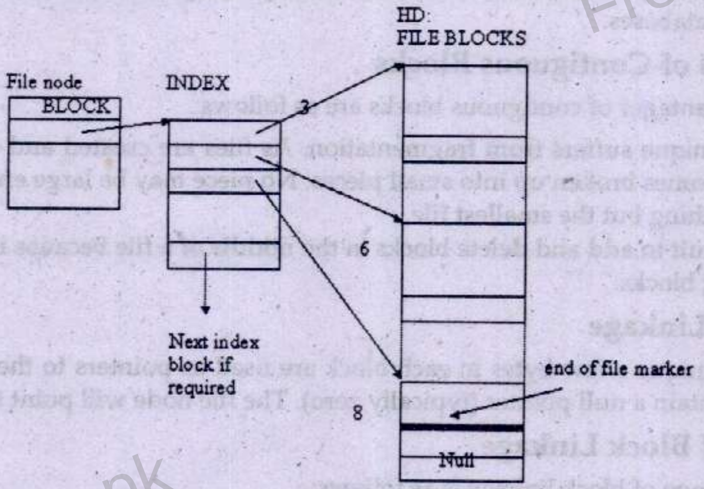
- Overhead needed to store the pointer of the next block. Some space is taken from each block to be reserved for the pointer.
- A large number of disk accesses are needed to find the end of the file. (Sometimes a pointer to the last block is stored in the file node).

- Access to the file is necessarily sequential since all blocks can be accessed only passing down the chain.
- Damage to one block can result in great data loss as reference to the following blocks is lost.


HD:
FILE BLOCKS
File node
3
6
End of file marker
8
Null

9.12.2.3 Index Blocks

In this technique the linkage pointers for each file are stored in a separate index block on disk. If the file is large then several index blocks may be needed, each one chained to the other. The file node will point to the first index block in the chain:


HD:
FILE BLOCKS
File node
BLOCK
INDEX
3
6
Next index block if required
end of file marker
8
Null

Advantages of Index Blocks

The advantages of index blocks are as follows:

- Fragmentation is eliminated since now any block can be reserved for the file.
- File can be accessed sequentially since any block can be accessed at random by simply specifying the name of the file and the offset in the index block.

Disadvantages of Index Blocks

The advantages of index blocks are as follows:

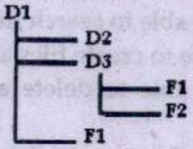
- Overhead needed to store the pointers of the blocks in the index block. Some space will always be lost from each index block.

- Addition or deletion of middle blocks will result in rearrangements of pointers in index blocks. This can be a lengthy operation.
- Damage to index blocks result in great data loss. This is often avoided by storing multiple copies of the index blocks in different areas of the disk. This is a substantial overhead but it provides a good solution.

9.13 Directories and Names

File system is used to organize files and other file system objects by using directories. A directory or **folder** is a file system object that contains other file system objects. A directory is an object that contains the name of file system objects. The distinction between name and object is not very important in a system with one-to-one mapping beween names and objects. In some file systems, the objects can have multiple names or no name. The distiction become important in this situation.

The entries in directories determine full pathname or names associated with a file system object. Starting with root directory, the full pathname is constructed by concatenating the sequence of names. The directory D1 in the following picture contains the names D2, D3 and F1. The directory D2 contains no names. The directory D3 contains the names F1 and F2.



The above organization defines the following six full pathnames on Unix;

- /D1
- /D1/F1
- /D1/D2
- /D1/D3
- /D1/D3/F1
- /D1/D3/F2

In Windows and DOS, the names would be same. The difference is that front slash "/" will be replaced by back slash "\".

9.14 Pre-Process Root Directory

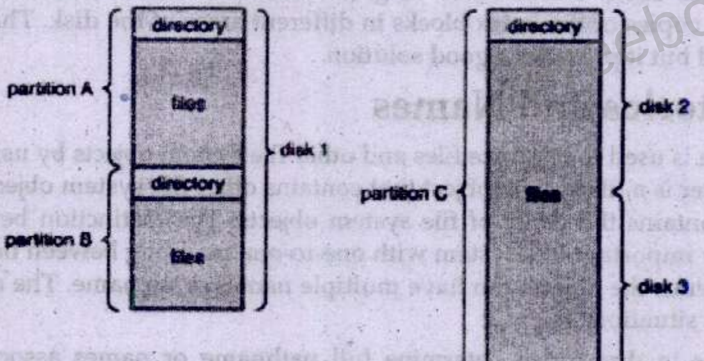
On some systems, it is possible to designate a directory that will serve as the root directory for a process. Once this designation occurs, only the subtree headed by that directory is visible to the process. On a system where file-sharing is desirable, each user's home directory could be made the root directory for all processes the user executes. The root of the file system would be the user's home directory to those processes. This capability may also be used by applications to provide additional security.

9.15 Directory Structure

In order to store many files, we need some organization in the way we store and reference them. Commonly, there are several levels of abstraction.

First is that disks are broken up into partitions. Each partition is logically treated as a separate hard drive. A disk can have many partitions. There is usually at least one partition on every hard drive.

Each partition (being a logical hard drive) has a *directory* that maintains a list of files in the directory.



9.15.1 Operations Performed on Directory

Just as with files, there are several basic operations on a directory:

- **Search for a file**: we need to be able to search the directory for a specific file.
- **Create a file**: We need to be able to create files and place the entry in the directory.
- **Delete a file**: We need to be able to delete a file and remove its entry from the directory.
- **List a directory**: We need to be able to get a list of files in the directory.
- **Rename a file:** We need to be able to rename files in the directory.
- **Traverse the file system:** We need some way (or ability) to traverse the directory. This is useful in backups, compression (zipping the entire directory), etc.

9.15.2 Single Level Directory

In a single-level directory system, all the files are placed in one directory. This is very common on single-user OS.

A single-level directory has significant limitations when the number of files increases or when there is more than one user. Since all files are in the same directory, they must have unique names. If there are two users who call their data file "test", then the unique-name rule is violated. Although file names are generally selected to reflect the content of the file, they are often quite limited in length.

Even with a single-user, as the number of files increases, it becomes difficult to remember the names of all the files to create files with unique names.

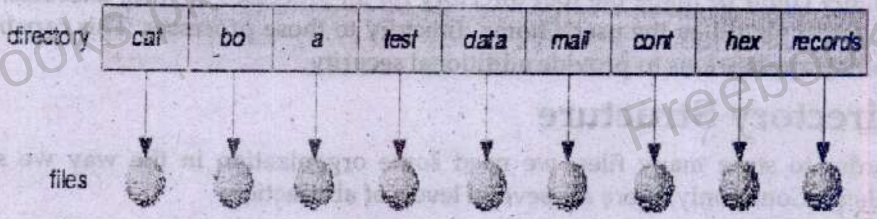


Figure: Single Level Director

9.15.3 Two-Level Directory

The next level of complexity is a two level directory structure. It solves some of the naming issues and perhaps has higher capacity. It can separate multiple users but cannot allow them shared access to files. It is not extendible in a general sense. Two level directories are not much better but they provide a bit more abstraction. For example, we can have several users having their own set of single-level directories

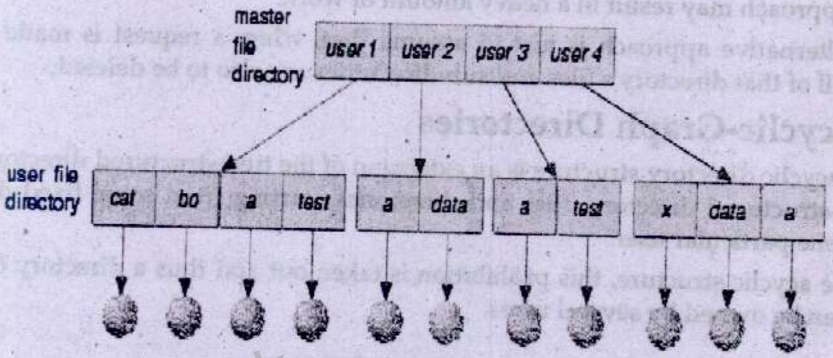

master file directory
user 1
user 2
user 3
user 4
user file directory
cat
bo
a
test
a
data
a
test
x
data
a

Figure: Two Level Directories

9.15.4 Tree-Structured Directory

In the tree-structured directory, the directories themselves are files. This leads to the possibility of having sub-directories that can contain files and sub-subdirectories.

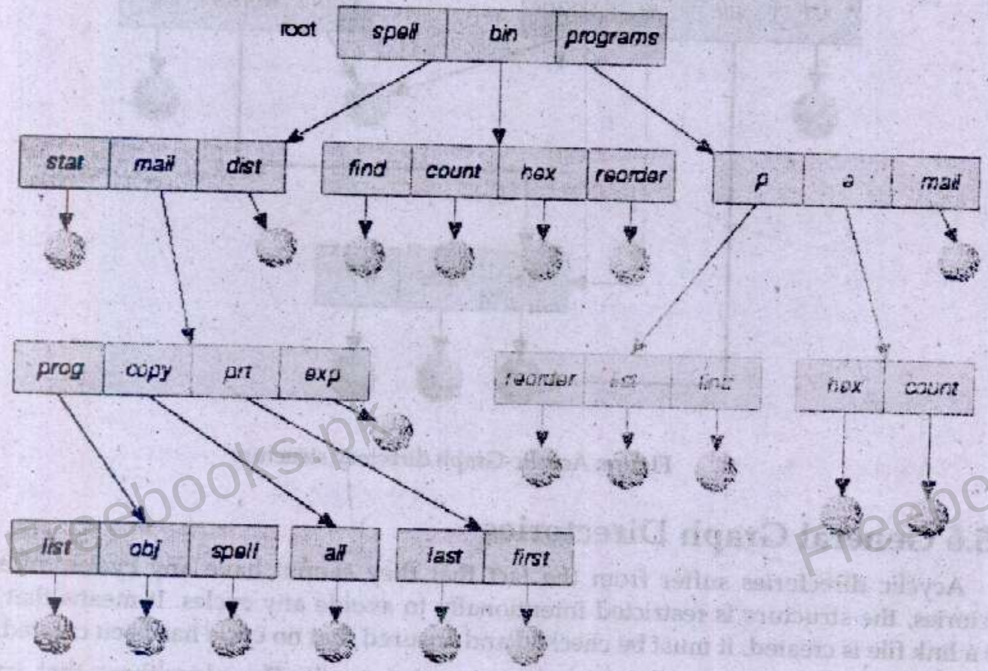

root
spell
bin
programs
stat
mail
dist
find
count
hex
reorder
p
e
mail
prog
copy
prt
exp
reorder
list
find
hex
count
list
obj
spell
all
last
first

Figure: Tree-structure directories

An interesting policy decision in a tree-structured directory structure is how to handle the deletion of a directory. If a directory is empty, its entry in its containing directory can simply be deleted. Suppose the directory to be deleted is not empty, but contains several files, or possibly sub-directories. Some systems will not delete a directory unless it is empty. The user must first delete all the files in a directory to delete that directory. If there are any sub-directories, this procedure must be applied recursively to them so that they can be deleted also. This approach may result in a heavy amount of work.

An alternative approach is just to assume that, when a request is made to delete a directory, all of that directory's files and sub-directories are also to be deleted.

9.15.5 Acyclic-Graph Directories

The acyclic directory structure is an extension of the tree-structured directory structure. In the tree-structured directory, files and directories starting from some fixed directory are owned by one particular user.

In the acyclic structure, this prohibition is taken out and thus a directory or file under directory can be owned by several users.

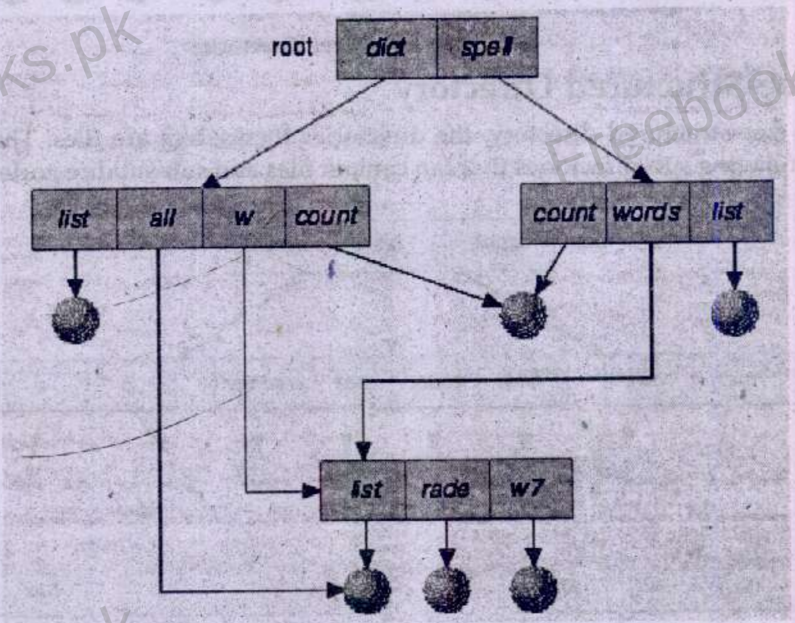


Figure: Acyclic-Graph directory structure

9.15.6 General Graph Directories

Acyclic directories suffer from the fact that they cannot have any cycles. In acyclic directories, the structure is restricted intentionally to avoide any cycles. It means that every time a link file is created, it must be checked and ensured that no cycle has been created.

General Graph directories allow cycles in the graph. The algorithms that traverse graphs and acyclic graphs are quite different. For a graph with possible cycles in it, we need to maintain a list of nodes we have already visited in order not to visit them again.

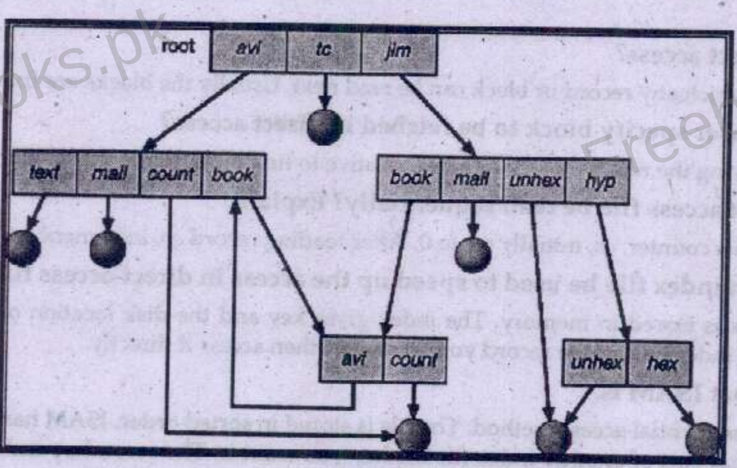


Figure: General Graph Directory

9.16 Directory Entries

Information about files is stored in directory entry. This information depends on the operating system. A directory entry contains information about the file ownership, location, size, access rights and times of events like last modication and creation etc.

Most operating systems require a file to be opened before it can be accessed. The open operation obtains a pointer to the files directory location. All subsequent references to the file occur via the pointer rather than file's name.

On some systems, the directory entry points to a separate file structure that contains information about the file's record. On Unix system, file information is stored in a structure called **inode**. The directory entry contains only the file's name and its inode number.

Review Questions

1. What is a file?

A named collection of related data defined by the creator, recorded on secondary Storage.

2. What do you know about directory or folder?

A directory or folder is a file system object that contains other file system objects. Actually, a directory is an object that contains the name of file system objects.

3. List advantages of operating system "knowing" and supporting many file types.

It can prevent users from making mistakes. It can make system convenient to use by doing various jobs automatically after one command.

4. List disadvantages of operating system "knowing" and supporting many file types.

The size of operating system becomes large. Every file type allowed must be defined. It hinders in creating new file types.

5. What is a sequential file?

A file that is read one record or block or parameter at a time in order. It is based on a tape model of a file.

6. What is direct access?

A file in which any record or block can be read next. Usually the blocks are fixed length.

7. How does user specify block to be fetched in direct access?

By specifying the relative block number, relative to first block in file that is block 0.

8. Can a direct access file be read sequentially? Explain.

Yes. Keep a counter, *cp*, initially set to 0. After reading record *cp*, increment *cp*.

9. How can an index file be used to speed up the access in direct-access files?

The index is stored in memory. The index gives key and the disk location of its corresponding record. Scan the index to find the record you want, and then access it directly.

10. Explain what ISAM is.

Indexed sequential access method. The file is stored in sorted order. ISAM has a master index file indicating in what part of another index file the key you want is. The secondary index points to the file records. In both cases, a binary search is used to locate a record.

11. List two types of system directories

a. Device directory: It describes physical properties of files.

b. File directory: It gives logical properties of the files.

12. List operations to be performed on directories.

Search for a file, create a file, delete a file, list a directory, rename a file, and traverse the file system.

13. List disadvantages of using a single directory.

Users have no privacy. Users must be careful in choosing file names, to avoid names used by others. Users may destroy each others' work.

14. What is the MFD? UFD? How are they related?

MFD is master-file directory. It points to the UFDs. UFD is user-file directory. It points to each of user's files.

15. What advantages are there to this two-level directory?

Users are isolated from each other. Users have more freedom in choosing file names.

16. What disadvantages are there to this two-level directory?

Two users who want to cooperate with each other are hampered in reaching each other's files. The system files are inaccessible.

17. How do we overcome the disadvantages of the two-level directory?

Provide links from one user directory to another creating path names. System files become available by letting the command interpreter search your directory first and then the system directory if file needed is not in first directory.

18. What is a file path name?

It is a list of directories, subdirectories and files traversed to reach a file from root directory.

19. Why would we want a subdirectory in our account?

It is used to group files into collections of similar nature and to protect certain groups of files from other users.

20. List steps you need to follow to delete a subdirectory in your account.

Delete all files in subdirectory. Change protection code to allow deletion and then delete the subdirectory. This procedure must be followed, starting with the deepest subdirectory.

21. What is an acyclic graph?

It is a tree that has been corrupted by links to other branches. It has no cyclic paths in it.

22. List ways to share files between directories in operating systems.

a. Copy file from one account into another.

b. Link directory entry of "copied" file to directory entry of original file.

c. Copy directory entry of file into account file is "copied" into.

23. What is a general graph?

A tree structure where links can go from one branch to a node earlier in the same branch or other branch allowing cycles.

24. What problems arise if the directory structure is a general graph?

Searching for a particular file may result in searching the same directory many times. Deletion of the file may result in the reference count to be nonzero even when no directories point to that file.

25. What is garbage collection?

It is the process of determining what file space is available and making it available for users.

26. How can we protect files on a single-user system?

a. Hide the disks.
b. Use file names that can't be read.
c. Backup disks.
d. On floppies, place a write-disable-tab on.

27. What might damage files?

Hardware errors, power surges, power failures, disk-head crashes, dirt, temperature, humidity, software bugs, fingerprints on magnetic material, bent disk or cover, vandalism by other users, storing diskettes near strong magnets which are found in CRTs, radio speakers, and so on.

28. List four ways systems might provide users to protect their files against other users.

a. Allowing user to use unprintable characters in naming files so other users can't determine the complete name.

b. Assigning password(s) to each file that must be given before access is allowed.

c. Assigning an access list, listing everyone who is allowed to use each file.

d. Assigning protection codes to each file, classifying users as system, owner, group and everyone else.

29. Explain the purpose of the open and close operations.

The *open* operation informs the system that the named file is about to become active.

The *close* operation informs the system that the named file is no longer in active use by the user who issued the close operation.

30. Give an example of an application in which data in a file should be accessed in the following order:

a. Sequentially
b. Randomly

Answer:

a. Print the content of the file.

b. Print the content of record *i*. This record can be found using hashing or index techniques.

31. In some systems, a subdirectory can be read and written by an authorized user, just as ordinary files can be.

a. Describe the protection problems that could arise.

b. Suggest a scheme for dealing with each of the protection problems you named in part a.

Answer:

a. One piece of information kept in a directory entry is file location. If a user could modify this location, then he could access other files defeating the access-protection scheme.

b. Do not allow user to directly write on subdirectory. Provide system operations to do so.

32. Explain the difference between a hard link and soft (or symbolic) link. What happens when the file you are linking to is deleted?

A hard link is a pointer to another file where as a symbolic link is a file that points to another file. When a hard link is created on UNIX, the file being linked to has its reference count increased by one. When the file or a hard link to it is deleted, the reference count is decreased by one. When the count reaches 0, the file is actually deleted. When a file is deleted, the symbolic link to it still exists but will (of course) not longer work.

33. List three ways of allocating storage, and give advantages of each.

a. Contiguous allocation. Fastest, if no changes are to be made. Also easiest for random-access files.

b. Linked allocation. No external fragmentation. File can grow without complications.

c. Indexed allocation. Supports direct access without external fragmentation.

34. What is contiguous allocation?

Allocation of a group of consecutive sectors for a single file.

35. What main difficulty occurs with contiguous allocation?

Finding space for a new file.

36. What is a "hole" in contiguous allocation method?

An unallocated segment of blocks.

37. Explain first-fit, best-fit, and worst-fit methods to allocate space for contiguous files.

First-fit: Scan available blocks of disk for successive free sectors. Use the first area found that has sufficient space. Do not scan beyond that point.

Best-fit: Search for smallest area large enough to place the file.

Worst-fit: Search for largest area in which to place the file.

38. What is external fragmentation in a system with contiguous files?

The disk has files scattered all over. Fragmentation occurs when there is enough empty space collectively for the next file but there is no single gap large enough for the entire file to fit in.

39. How can we overcome fragmentation?

We can use an allocation technique that does not result in fragmentation. We can move the files around on disk putting them closer together to get larger blocks of available sectors.

40. What is pre-allocation? Why do it?

Allocating space for a file before creating it allow expansion. This reserves space for a particular file so that other files can't grab it. The new file may initially use only a small portion of this space.

41. What is linked allocation, as detailed in text?

Directory contains pointers to first and last blocks of file. Each block of file (except last) has pointer to the next block.

42. Can linked allocation have external fragmentation or internal fragmentation?

External – no. Internal – Yes.

43. What is indexed allocation?

Each file has its own block of pointers to the sectors of the file.

44. Rank the allocation methods on speed.

Contiguous is fastest. Linked is slower because the disk head may have to move between accesses of file. Indexed is slowest unless the entire index can be kept in memory at all times. If not, then extra time must be used to access next block of file indexes.

10

I/O & DEVICE MANAGEMENT

Chapter Overview

10.1 I/O Management Module
10.2 Objectives
10.3 Buffering
10.4 Spooling
10.5 Handshaking
10.6 Interrupts
    10.6.1 Software Poll
    10.6.2 Vectored Interrupts
10.7 Interrupt Priorities
    10.7.1 Disable interrupts
10.8 Classes of Interrupts
10.9 I/O Devices
10.10 Storage hierarchy
10.11 Hardware Organization
    10.11.1 Single-Bus Architecture
    10.11.2 Hardware Organization
    10.11.3 I/O Control
    10.11.4 Ports and Memory-Mapped I/O
    10.11.5 Module Registers
    10.11.6 Busy Wait I/O
    10.11.7 Polled I/O
    10.11.8 Interrupt I/O
    10.11.9 Direct Memory Access
10.12 Software Organization
    10.12.1 Network I/O
    10.12.2 Logical I/O
    10.11.3 Buffering
    10.12.4 Caching
    10.12.5 Device Drivers
Review Questions

10.1 I/O Management Module

The I/O management module in the operating system provides a means by which a process can communicate with the outside world i.e. the mechanism for input and output information. Traditionally the I/O system is regarded as the most difficult to implement because of the different number of peripherals that can be used with a different configuration.

External devices mainly differ in the following ways:

1. Speed

There may be a difference between the data transfer rates of various devices. A magnetic disk may be able to transfer 10^12 characters per second. A keyboard may transfer only a few characters per second depending on the typist.

2. Amount of Data Transferred

Data may be transferred in units of characters, words, bytes, blocks or records, according to the peripheral used.

3. Data Representation

Data may be encoded in different ways on different I/O media. Even on a single medium, such as magnetic tape, several different codes may be employed.

4. Permissible Operations

Devices differ in the kind of operation they can perform. One example is the distinction between input and output. One ability is to rewind a magnetic tape but not printer paper.

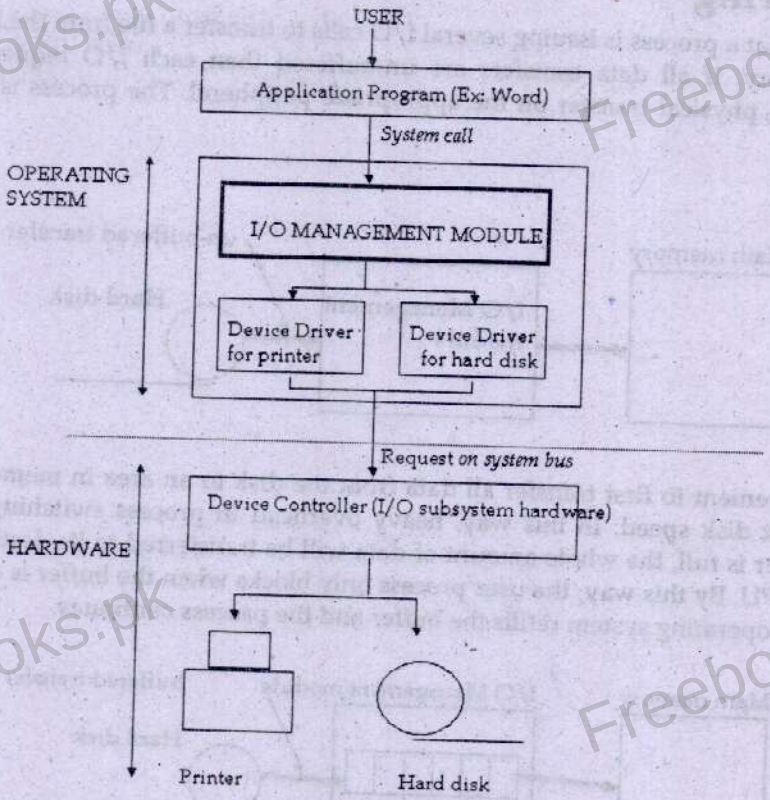
5. Error Conditions

Failure to complete a data transfer may have various causes. A hardware detected data error or a printer that has run out of paper depending on the peripheral being used.

It would make the operating system too large if it caters for every specific detail. So I/0 system must isolate the device characteristics and treat all devices uniformly possibly.

1. The I/O management module provides a common interface for user applications to interact with external devices. For example to output a file to some device the interface may provide a common function called *OUTPUTFILE*.
2. It is up to the I/O management module to redirect the call from the user to the appropriate device driver. A **device driver** is a software module that manages the communication with a specific I/O device or type of device. It is the task of the device driver to convert the logical requests from the user into specific commands directed to the device itself.
3. For example on receiving the *OUTPUTFILE* command mentioned above, the device driver of a floppy drive would first check for the presence of a disk in the drive, locate the file via the filing system, position the heads on the floppy and the starts transferring data to the device controller.

Below is a diagram that shows how the I/O module fits in a system that separates the user at one end from the physical devices at the other end:


USER
Application Program (Ex: Word)
System call
OPERATING SYSTEM
I/O MANAGEMENT MODULE
Device Driver for printer
Device Driver for hard disk
Request on system bus
Device Controller (I/O subsystem hardware)
HARDWARE
Printer
Hard disk

10.2 Objectives

The objectives of the I/O Management module are as follows:

1. Generality and Device Independence

I/O devices are typically quite complex mechanically and electronically. Much of this complexity is related to the electronic engineering and is of no interest to the user or the programmer. The average user is not aware of the complexities of positioning the heads on a disk drive, reading the signal from the disk surface, waiting for the required sector to rotate into position etc.

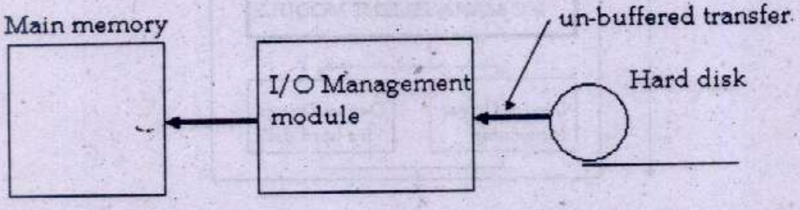
Users and programmers may be unaware of this complexity. The I/O management module must try to treat all external devices uniformly. This is achieved by **virtual device**. A virtual device is a special kind of file that is associated with a particular external device. Users create a virtual device of a given type, and operating system associates a physical device with it when the device is required for the first time. All virtual devices in a system are stored in a linked list.

2. Efficiency

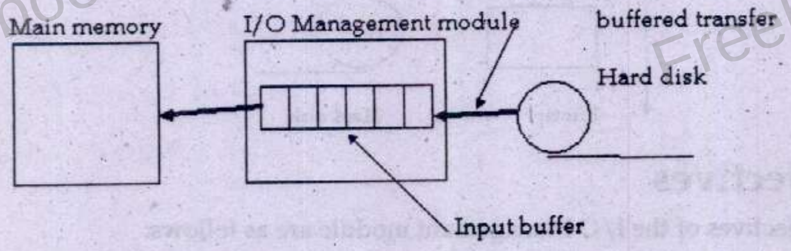
Perhaps the most significant characteristic of the I/O system is the speed disparity between it and the processor. I/O devices involve mechanical operations. They cannot compete with the microsecond or nanosecond speed of the processor and memory. The I/O management module must try to minimize the disparity by the use of techniques like **buffering** and **spooling**.

10.3 Buffering

Assume that a process is issuing several I/O calls to transfer a file from the hard disk to the main memory. If all data transfers are un-buffered then each I/O request from the process causes a physical transfer on the appropriate peripheral. The process is blocked on each request.



It is convenient to first transfer all data from the disk to an area in memory called an **input buffer** at disk speed. In this way, heavy overhead in process switching is avoided. When the buffer is full, the whole amount of data will be transferred to its destination at the speed of the CPU. By this way, the user process only blocks when the buffer is empty. When this occurs the operating system refills the buffer and the process continues.

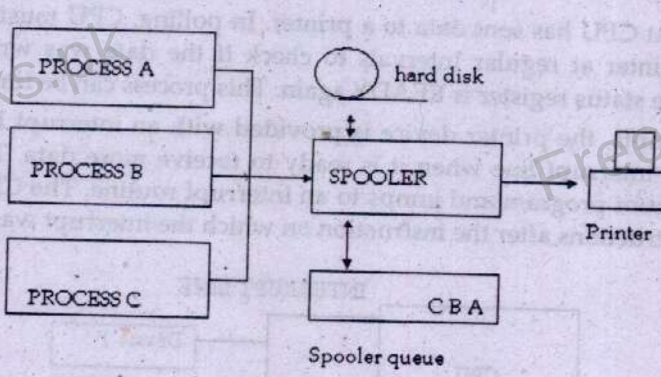


In the same way, the output transfer from a process to an external device are directed to an **output buffer**.

10.4 Spooling

In spooling, a high-speed device like a disk is placed between a running program and a low-speed I/O device. It prevents the program execution time from being limited by the speed of I/O device. One of the most common types of spooling is print spooling. A program does not send its output to a relatively slow mechanical printing device. The output is placed in a disk file or spool file. The program can run at the speed of the disk and therefore finish more quickly. The printer reads the data from the spool file when it becomes available. The disk is much faster than the printer. The spooling program can drive the printer at top speed while the original user program proceeds with new activities in parallel with the spooling.

Spooling involves all processes directing any output data to an un-sharable device to a component of the I/O management module called the **spooler**.



The main functions of spooler are as follows:

- It receives and stores the output on hard disk and enters the job in a spool queue.
- When the un-sharable device becomes available, it picks the first job from the queue and transfers its data from the hard disk to the un-sharable device.

Advantages of Spooling

The advantages of spooling are as follows:

- Processes are not suspended for a long time.
- It can produce multiple copies of the output without running the process again.

Disadvantages of Spooling:

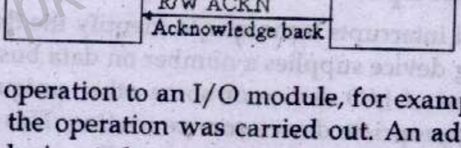
The disadvantages of spooling are as follows:

- Need large amounts of disk space.
- Increase disk traffic.
- Not practical for real-time environment, because results are produce at a later time.

10.5 Handshaking

Handshaking is a technique used for controlling the movement of data between system components. For example, before an I/O module sends data to the CPU, it must make sure that the CPU is ready to receive that data. On the other hand if the CPU wants to send data to a particular device it must make sure the data was received before sending other data. Almost all system components have control lines used for handshaking.

In general these are called **Acknowledge Control Lines**:



After a read or write operation to an I/O module, for example, the R/W ACKN line is raised to tell the CPU that the operation was carried out. An advantage of handshaking is that it allows fast and slow devices to be accommodated on the same system bus.

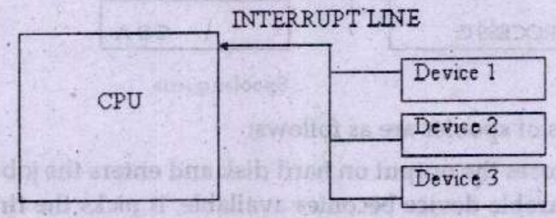
10.6 Interrupts

CPU uses the following techniques to know if a device needs servicing:

1. Polling
2. Interrupts.

Suppose that CPU has sent data to a printer. In polling, CPU must examine the status register of the printer at regular intervals to check if the data was written. The checking continues until the status register is READY again. This process can be time consuming.

With interrupts, the printer device is provided with an interrupt line on the bus. The printer raises the interrupt line when it is ready to receive more data. The CPU then stops executing the current program and jumps to an interrupt routine. The CPU then jumps back to execute the instructions after the instruction on which the interrupt was executed.



The following things must be done each time an interrupt is generated by a device:

1. CPU finishes the execution of current instruction before responding to the interrupt.
2. The location of the next instruction in memory must be saved. Then the program can be re-entered when the interrupt servicing program is finished.
3. The device that generated the interrupt must be identified. An acknowledgment is given so that the device will lower the interrupt line.
4. The CPU must jump to the program that will service the interrupt.
5. The CPU must fetch the saved instruction and continue the original program when the interrupt has been serviced.

There are two important issues in implementing interrupts. First, CPU needs to decide which device issued the interrupt as there will be multiple I/O modules in a system. Secondly, CPU needs to decide which interrupt should be processed if multiple interrupts have occurred.

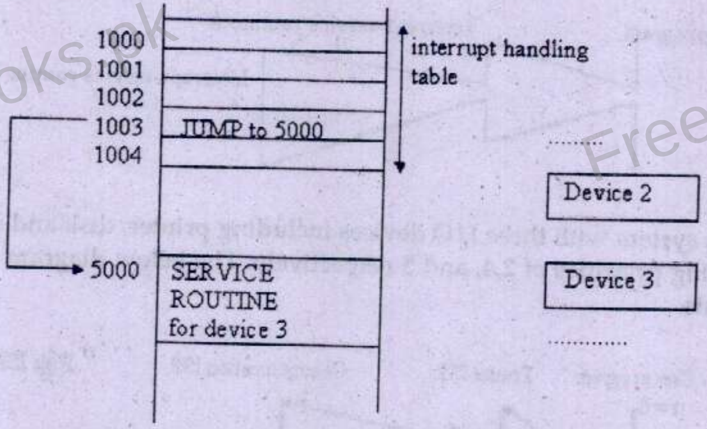
10.6.1 Software Poll

CPU can use software poll technique to identify the device that caused the interrupt. CPU executes an interrupt-service routine when it detects an interrupt. This routine poll each I/O module to determine the module that caused the interrupt. CPU reads the status register of each I/O module in turn to identify the interrupting module. When the correct device is identified, CPU executes a device-service routine specific to that device. The disadvantage of software poll technique is that it is time consuming.

10.6.2 Vectored Interrupts

CPU can use vectored interrupts technique to identify the device that caused interrupt. In this case, the interrupting device supplies a number on data bus. The number is referred to as a vector. It is the address of I/O device or some other unique identifier. CPU uses the vector as a pointer to the appropriate device-service routine. This avoids the need to execute a general interrupt-service routine first as in software poll.

In general, CPU will have an interrupt-handling table in memory beginning at a known based address. Thus if device number 3 wants to interrupt the CPU, it raises the interrupt line and puts 3 on the data bus. CPU will add this number on the data bus to the base address to find the next instruction to execute. The instruction in the interrupt-handling table will be a JUMP to the service-routine for device 3.


1000
1001
1002
1003
1004
JUMP to 5000
interrupt handling table
5000
SERVICE ROUTINE for device 3
Device 2
Device 3

10.7 Interrupt Priorities

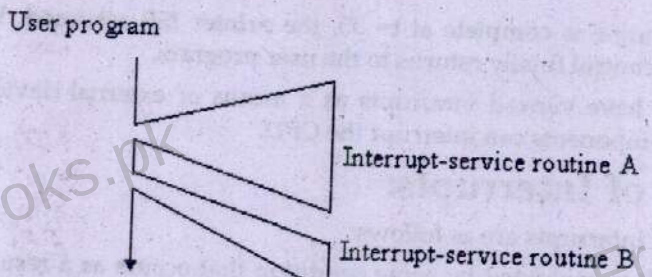
Suppose that multiple interrupts can occur. For example a program may be receiving data from a communication line and printing results. The printer will generate an interrupt every time that it completes a print operation. The communication line will generate an interrupt every time a unit of data arrives. In any case, it is possible for a communication interrupt to occur while a printer interrupt is being processed.

Two approaches that can be used are as follows:

1. Disable Interrupts
2. Define Priorities

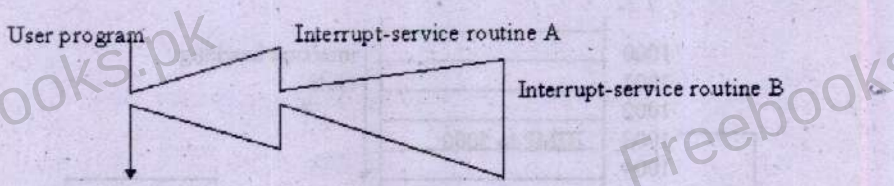
10.7.1 Disable Interrupts

It means that CPU can and will ignore interrupt request signal. If an interrupt occurs during this time, it generally remains pending and will be checked by the CPU after CPU has enabled interrupts. The interrupts are disabled immediately when a user program is executing and an interrupt occurs. After the interrupt-service routine completes, interrupts are enabled before resuming the user program. CPU checks to see if additional interrupts have occurred. Interrupts are handled in strict sequential order:

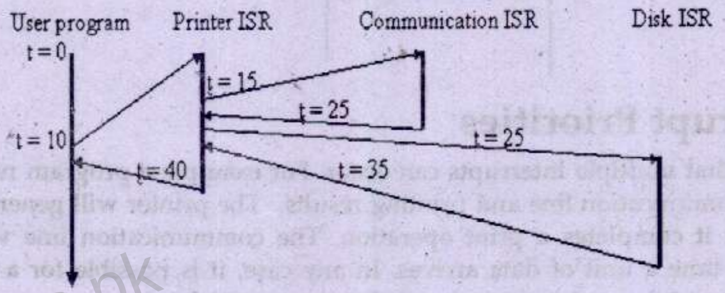

User program
Interrupt-service routine A
Interrupt-service routine B

10.7.2 Define Priorities

A second approach is to define priorities for interrupts and to allow an interrupt of higher priority to cause a lower-priority interrupt-service routine to be itself interrupted as shown below :



Consider a system with three I/O devices including printer, disk and a communication line with increasing priorities of 2,4, and 5 respectively. The below diagram shows a possible sequence of events:



A user program begins at time t = 0. A printer interrupt occurs at t = 10. The user information is placed on system stack and execution continues at the printer interrupt-service routine(ISR). While this routine is still executing, a communication interrupt occurs at t = 15. Since the communication line has a higher priority than the printer, the interrupt is honored.

The printer ISR is interrupted. Its state is pushed onto the stack and execution continues at the communication ISR. While this routine is executing, a disk interrupt occurs at t = 20. Since this interrupt is of lower priority, it is simply held and the communication ISR runs to completion.

When the communication ISR is complete at t = 25, the previous CPU state is restored that is the execution of the printer ISR. However, before even a single instruction in that routine can be executed, CPU honors higher-priority disk interrupt and control transfers to the disk ISR.

When that routine is complete at t= 35, the printer ISR resumed. When that routine completes at t = 40, control finally returns to the user program.

Although we have viewed interrupts as a means of external devices to notify CPU, almost all system components can interrupt the CPU.

10.8 Classes of Interrupts:

The classes of interrupts are as follows:

Program	It is generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero or attempt to execute an illegal machine instruction.
Timer	It is generated by a timer within the CPU. This allows the operating system to perform certain functions on a regular basis.
I/O	It is generated by the I/O to signal normal completion of an operation or an error.

Hardware Failure It generated by hardware failure.

10.9 I/O Devices

I/O devices can be roughly divided into two categories known as **block devices** and **character devices**. A block device is one that stores information in fixes-size blocks, each one with its own address. Common block sizes range from 128 bytes to 1024 bytes. The essential property of a block device is that it is possible to read or write each block independently of all the other ones. In other words, at any instant, the program can read or write any of the blocks. Disks are block devices.

Some devices are block addressable and some are not. The boundary between these devices is not well defined. A disk is a block addressable device. It is always possible to seek another cylinder and wait for the required block to rotate under the head.

Consider the magnetic tape containing blocks of 1K bytes. If the tape drive is given a command to read block N, it can rewind the tape and go forward until it comes to block N. It may or may not be possible to rewrite one block in the middle of a tape.

The other type of I/O device is the character device. A character device delivers or accepts a stream of character without regard to any block structure. It is not addressable and does not have any seek operation. The terminals, line printers, paper tapes, punched cards, network interface, mice and most other devices are character devices.

Some devices just do not fit in this classification. However, the model of block and character devices is general enough that it can be used as a basis for making the I/O system device independent.

10.10 Storage Hierarchy

Main memory is usually located on chips inside the system unit. Two types of memories are **random-access memory (RAM)** and **read-only memory (ROM)**.

The instructions and the data are kept in RAM during execution. RAM is not a permanent storage place for information. It is active only when computer is on. If computer is switched off, the information is deleted from RAM.

ROM is the memory where the information can only be read. When the computer is turned off, the information is not deleted from ROM. Information to ROM is written by vendor. The size of the main memory is measured in megabytes.

The information stored on a disk is not deleted when the computer is turned off. Information stored on the disks is moved in and out of RAM. There are two kinds of disks: **hard disk and floppy disk**.

The main memory and the floppy disk have less storage capacity than the hard disk. The hard disk can write and read information to and from the main memory much faster than a floppy disk. The access speed of main memory is also much faster than a hard disk.

Most programs are stored on a disk until loaded into memory. The proper management of disk storage is of central importance to a computer system.

The storage in a computer system can be organized in a hierarchy according to speed or cost. The higher levels are expensive but very fast. As we move down the hierarchy, the cost per bit decreases, while the access time increases and the amount of storage at each level increases.

Caching is an important principle of computer systems, both in hardware and software. Information is normally stored in some storage system like main memory. As it is used, it is copied into a faster storage system (the cache) temporarily. When some information is needed, we first check if it is in the cache. If it is, we use the information directly from the cache. If not, we use the information from the main storage system, putting a copy in the cache.

Since cache have limited size, cache management is an important design problem. Careful selection of cache size and a replacement policy can mean that 80 to 90 percent of all accesses are in the cache, resulting in very high performance.

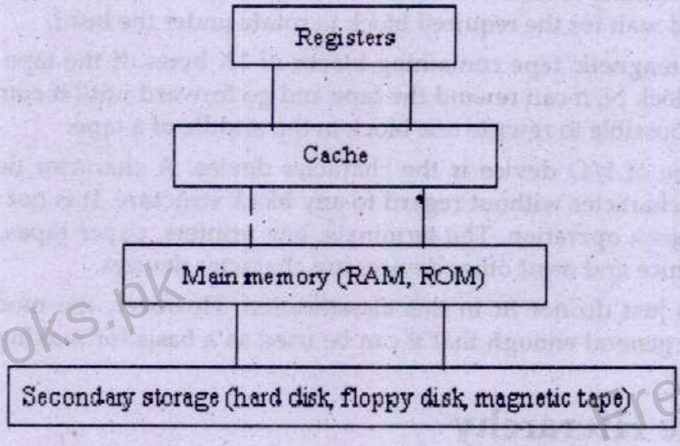


Figure: Storage hierarchy

The programmer (or compiler) implements the register allocation and replacement algorithms to decide what information to keep in registers, and what to keep in main memory. The movement of information between levels of a storage hierarchy may be either explicit or implicit.

10.11 Hardware I/O Organization

I/O devices, memory and CPU communicate with each other by using one or more communication buses.

10.11.1 Single-Bus Architecture

The simplest computer uses a single communication bus. All devices are attached with this bus. Only two devices can communicate with each other at one time. Some protocol is required to control the communication. Time is broken into **clock cycles**. Clock cycle is the time required to send a piece of information on bus. A special device **bus arbitor** decides which device will communicate in the next cycle. The selected device can communicate with any device if the device is connected to the system.

Usually, the bus is allocated to CPU to communicate with memory. CPU utilizes the bus more efficiently due to its high speed. I/O devices require the buss less frequently than CPU. But the communication needs of I/O devices are more time-critical. So the requests for the bus from I/O devices are given high priority than CPU. The process of taking the bus away from CPU and giving it to I/O device is called **cycle stealing**.

10.11.2 Multiple-Buses Architecture

Multiple buses are used for the following reasons:

Parallelism

Multiple communications are possible at the same time with multiple buses. For example, CPU can communicate with a port on one bus and a disk can communicate with memory on the other bus. The advantages of multiple buses are still limited. The reason is that most communications use either memory or CPU.

Performance Tuning

Multiple buses enhances the performance of the system. Latest computer architectures use three types of uses:

- Peripheral Component Interconnect (PCI) bus is used to connect devices.
- Special memory bus is used for optimized communication between CPU & memory.
- Industry Standard Architecture (ISA) bus is connected to PCI bus to provide backward compatibility for older ISA devices.

10.11.3 I/O Control

In the simple I/O control model, CPU communicates with I/O devices directly. CPU controls all details of device operation. This type of communication is only used in specialized, microprocessor-controlled, embedded systems.

In PC model, CPU communicates with a **device controller**. A signal from CPU to device controller may start a read operation from a device. The detailed commands of read operation are performed by device controller. CPU can perform other processing in the mean time. Each I/O controller is designed to handle a particular kind of device.

Some sophisticated systems use an additional level of I/O control. CPU communicates with an I/O channel. This channel communicates with an I/O control unit. I/O channels are more sophisticated than I/O controllers and may have their own CPU. They execute channel programs loaded in the main memory or channel's memory.

10.11.4 Ports and Memory-Mapped I/O

CPU must communicate with I/O module to perform any I/O operation. An I/O module contains one or more registers. The communication is done by reading or writing to these registers. The I/O module sets the values of registers to communicate with CPU.

CPU access the module registers in two ways:

1. Port I/O

In port I/O, a limited number of instructions provide I/O capability. One instruction operands is an I/O port number. I/O operation is applied to I/O module identified by that port.

2. Memory-Mapped I/O

In memory-mapped I/O, any instruction that accesses memory may access an I/O module. A part of memory space is allocated to I/O modules. It is a disadvantages in system with limited address space.

10.11.5 Module Registers

Different I/O modules have different types and number of registers. A simple module for input device may have only two registers; one for data and one for control information. The data register stores the vlaue and control register contains bits to command I/O module. The bits may be used to enable the device, reset the device or begin a read operation etc. Some bits are used to signal operation completion, device ready or error conditions etc. The modules for complicated devices contains more registers.

10.11.6 Busy Wait I/O

Busy wait is a protocol for communication between I/O module and CPU. CPU issues a signal to I/O module to start I/O operation. In output operation, data is transferred to data buffer of the module. CPU repeatedly checks for the completion of the command. When it is completed, status information is checked to verify the success of the operation. In input operation, data is transferred from the module buffer. Busy wait I/O is inefficient because processes can make no headway while CPU waits for the completion of I/O operation.

10.11.7 Polled I/O

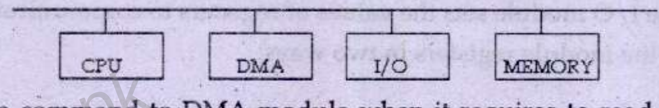
In polled I/O, CPU continues to execute different processes in the system after initiating I/O operation. CPU checks for the completion of I/O operation after brief intervals. Polling is more efficient. The disadvantage is that the processes are interrupted when CPU checks for the completion of I/O operation.

10.11.8 Interrupt I/O

An interrupt can be sent to CPU when I/O operation is completed. In this way, CPU continues to execute other process until it gets an interrupt from I/O module. It provides parallelism but adds complexity in the code.

10.11.9 Direct Memory Access (DMA)

Direct Memory Access (DMA) is a technique that is used to send data directly from a device to the memory. CPU is not involved in this data transfer that results in speeding up the overall computer operation. Usually a specified portion of memory is designated for direct memory access. DMA is required when a large volume of data is moved. It involves an additional module on the system bus.



CPU issues a command to DMA module when it requires to read or write a block of data. It sends the following information to DMA module:

1. Types of operation required such as read or write.
2. The address of I/O device involved.
3. The starting locations in memory to read from or write to.
4. The number of words to be read or written.

The CPU then continues with other work. It has delegated I/O operation to the DMA module. DMA module will then transfer the entire block of data one word at a time directly to or from memory without going through CPU. DMA module sends an interrupt signal to CPU when the transfer is complete. CPU is only involved at the beginning and at the end of data transfer. DMA works while CPU is idle and is not using the system bus. It is called **cycling stealing.**

10.12 Software Organization

Operating system implements supervisor calls for processes to request device input and output services. The same set primitive may be available for all devices.

Some characteristics of devices affect the supervisor call that are as follows:

1. Device Speed

The speed of device I/O can be slow. Some operating systems provide the facility to issue nonblocking I/O operations that can be handled as follows:

Using data received and buffered before the system call, the return can be immediate using the data in the input buffer. The return value indicates the amount of supplied data.

The process can be notified at the completion of input operation. It can be done by sending a signal to process. Signal is sent by setting the value of a process variable or by providing a special system call.

Notification can be ignored in nonblocking outputs. The reason is that it is not relevant to the application when the data is output by the device.

2. Mutual Exclusion

Operating system provides the following protection for nonsharable devices:

- It may allow only one process to open the device at one time.
- It may assign the ownership of the device to the user who opened it and may deny access to other users.
- A spooling scheme may be used to handle the problem at application level.
- It may do nothing and leave the problem to the user.

3. Unidirectional Devices

Some devices are unidirectional i.e. input-only or oupt-only. The devices can only be used for specific purpose. The devices must be opened before I/O can be performed on them. The open call can include a parameter that indicates the type of access. An attempt to open a device for wrong kind of I/O should not be allowed.

10.12.1 Network I/O

Some operating systems provide separate mechanism for network devices. These operating systems provide the support for network protocols for complex network communication. A popular network I/O abstraction is **sockets**.

Socket is connected to your computer on one side and to a remote computer on the other side. Applications create sockets, establish connections to sockets on remote machines, and send and receive messages on established connections.

10.12.2 Logical I/O

Logical I/O level uses semi-device-independent kernel I/O objects. The objects are not always totally device-independent. I/O requests from other modules of operating system access the capabilities of this level.

Two commonly device types are block devices and character devices. A disk is a block device. I/O operations transfer the number of disk blocks or sectors. Character devices like printer and modem transfer single character or byte at a time.

10.12.3 Buffering

Logical I/O provides buffering for I/O data for the following reasons:

- I/O operation can occur at any time. If no process requests input from keyboard, the buffers are used to store data typed by the user.
- Block devices perform I/O in blocks. Buffers package the requests in block-size units.
- In DMA transfers, buffers are used as destination or source of transfer.
- In asynchronous write operation, data is copied in buffer and application continues to execute.

10.12.4 Caching

Cache is a fast storage device to hold a copy the data stored on a slower device. If the information to be read from a slower device is present in cache, the request cn be executed faster. I/O caches storage device data in main memory. Systems that implement caches create a buffer to be used as cache.

10.12.5 Device Drivers

I/O objects generated in the logical I/O software are passed to device drivers that are built for the needs of different devices. They operate on one or more related devices. Operating system may require a set of functions implemented for that device as follows:

- **open:** Perform startup function before the use of device.
- **close:** Perform any function to shut down the device.
- **schedule:** Perform scheduling on I/O request.
- **startio:** Start next I/O if the device is not busy.
- **interrupt:** The routine executed when the device sends an interrupt to CPU.
- **ioctl:** Implements any special functions that can be applied to this device.

Review Questions

1. How CPU read from memory or I/O?

CPU puts the address on the address bus and activate proper control line (Memory Read or I/O Read) and wait until data become available on the data bus from memory or I/O. Memory or I/O puts the data onto the data bus as soon as it has recognized its address. Then CPU reads data from data bus.

2. How CPU write to memory or I/O?

CPU puts the data on to the data bus and puts the address on to the address bus at the same time. The memory or I/O is activated by proper control line then recognizes its address and read the content of the data bus.

3. How is an interrupt executed?

The I/O driver sends a signal through a special interrupt line to the CPU when it has finished with an I/O request.

4. Why we say that modern operating systems are interrupt driven?

If there are no processes to execute, no I/O devices to service and no users to respond, an operating system will sit quietly. It will wait for something to happen. Events are almost always signaled by the occurrence of an interrupt or a trap.

5. In what ways do systems treat slow and fast devices differently?

For slow devices, each character transferred causes an interrupt. For fast devices, each block of characters transferred causes an interrupt.

6. What is DMA?

Direct Memory Access allows transfers of blocks of data between memory and I/O devices without intervention of CPU. It makes transfers in between CPU instruction executions called interleaving.

7. Differentiate DMA device controllers and channel controller?

A channel controller can do everything a DMA device controller can do. Additionally, it can execute arbitrary programs. It can relieve CPU of the burden of executing much of the I/O code of the operating system and/or the database management system. Channel controllers cost a lot. They usually have their own path to memory to minimize interference with the main CPU(s).

8. What is cycle stealing? Why are there cycles to steal?

When DMA device transfers data to or from memory, it will use the same bus as CPU would use to access memory. If the CPU wants to use the bus at the same time as a DMA device, the CPU will stall for a cycle, since the DMA device has the higher priority. This is necessary to prevent overruns with small DMA buffers. Modern CPUs have caches that satisfy most memory references without having to go to main memory through bus. DMA will therefore have much less impact on them.

9. Why caches are useful?

Caches are useful when two or more components need to exchange data and the components perform transfers at differing speeds. Caches solve the transfer problem by providing a buffer of intermediate speed between the components. If the fast device finds the data it needs in the cache, it need not wait for the slower device. The data in the cache must be kept consistent with the data in the components. If a component has a data value change and the data is also in the cache, the cache must also be updated. This is especially a problem on multiprocessor systems where more than one processes may be accessing a datum. A component may be eliminated by an equal-sized cache, but only if:

10. What is a disk controller?

A separate board installed in the computer chassis to control and interact with each disk drive attached to it.

11 OPERATING SYSTEM SECURITY

Chapter Overview

11.1 Security
11.2 Security Problem
11.3 Authentication
    11.3.1 Passwords
    11.3.2 Password Vulnerabilities
    11.3.3 Encrypted Passwords
    11.3.4 One Time Passwords
11.4 Prevention
11.5 Detection
11.6 Correction
11.7 Identification
11.8 Program Threats
    11.8.1 Trojan Horse
    11.8.2 Trap Doors
11.9 System Threats
    11.9.1 Worms
    11.9.2 Viruses
11.10 Threat Monitoring
11.11 System Security

Review Questions

11.1 Security

The need for security in computing systems is growing rapidly. This is mainly due to the critical services that are dependent on interconnected computing systems. Computer networks are now a necessity for national infrastructure components such as electric power, telecommunications and transportation systems.

Security and protection are firmly internal problems. In a computer system, the way we provide controlled access to programs and data stored in a computer is by security. Security system requires an adequate protection system. It must also consider the external environment in which the system operates. The Internet has increased the public's concern for security immensely. Security is the primary concern for businesses, especially when they are using the Internet for commerce and maintaining both relationships and communication.

Information such as data and code stored in a system needs to be protected from unauthorized access, malicious destruction or alteration, and accidental inconsistency.

11.2 Security Problem

Operating systems provide various mechanisms to protect data and programs. A system is secure if and when the resources are used and accessed as intended under all circumstances. Security is never assured to be one hundred percent. Security violations are either maliciously intentional or accidental.

Some examples of malicious accessing are as follows:

- Unauthorized reading of data
- Unauthorized modification of data
- Unauthorized destruction of data

Security measures must be taken to both physical and human levels. For the physical level, computer system's sites need to be physically secured from intruders. On the human level, users need carefully be given authority because the user may give access to an intruder.

Carelessness at a high level of security (physical or human) will lead to the need of strict low-level (operating system) security measures. A considerable effort is required to prevent any form of accident or fraud. On the other hand, the system hardware must provide protection to implement security features. Operating System security is implemented in various levels.

11.3 Authentication

Authentication is the major security problem for operating systems. Protection of the system depends on identifying programs and processes that are executing. Each user using the system should also be identified. A user can be identified by:

- User possession, using such as a key or card
- User knowledge, using an identifier or password
- And user attributes, such as fingerprint, retina pattern, or signature

11.3.1 Passwords

The most used form of security and user validation is by testing the user's knowledge and using a password. Users are normally asked for two things, their user ID, account name or login name and then their password. If the user password is supplied and the password stored in the system match, then the system believes the user is valid.

11.3.2 Password Vulnerabilities

Passwords do not provide total security and have problems. They are common because they are easy to use and understand. The problems with passwords are keeping them secretly. Passwords can be guessed, accidentally exposed, and maliciously transferred from an authorized user to an unauthorized user.

- An **intruder**, either a human or a program, can try and guess a password by entering obvious information about the user or by **brute force**. If the intruder gets information such as name, spouse name, etc. it can be easy to guess correctly.
- Sometimes there are limitations to what the password can consist of. For example, if it can only be a certain number of characters, it becomes easier to guess the password. A program can be written to produce all the possible numbers of a restricted amount of digits and can find the password in seconds.
- Another failure to keeping a password secret is due to visual or electronic monitoring. An intruder can look at the keyboard when the user enters password and steal it by watching what keys they press carefully.
- Sometimes, the user willingly gives password to his friends for some help. That can misuse the trust and use it as an intruder.

Some preventions from password vulnerabilities are as follows:

- Use a longer password
- Password should be a combination of both numbers and characters
- Passwords should be case sensitive.

11.3.3 Encrypted Passwords

Encryption is used for more security. A function is used to encode all passwords, and only the encoded passwords are stored. Once a user gives a password, it is directly encoded and then compared to the stored encoded password.

One problem with this method of password testing is that the system will no longer have control over the passwords. The encrypted password exists in some file. Anyone with access to that file can simply run encryption routines very fast against it, to encrypt each word in the file, and then compare it's results to for example a dictionary. Now if the password exists in the dictionary, then the password is easily cracked.

11.3.4 One Time Passwords

One way to solve password security problems is to use a set of paired passwords. The user is challenged by system and he must respond with the correct answer. First, the system randomly selects and presents one part of the password of a password pair. It then asks the user to supply the second part. If the parts make the pair then the user is valid.

In this one-time password case, the password is different in each instance. Therefore, anyone attempting to steal the password and reuse it will fail.

11.4 Prevention

The basic purpose of a security system is to prevent intruders from penetrating the system security. Preventing measures include the following

- Passwords should be used after a series of quality checks. Software are available that can check a password for length and diversity of characters etc.
- Passwords should be changed at periodic intervals.

- Data should be encrypted when it is transmitted or when it is stored.
- The system entry points should be reduced.
- An internal firewall should be implemented to deny network access from remote locations.

11.5 Detection

If a break-in occurs, it should be detected as soon as possible. Effective detection measures also discourage intrusion attempts.

- Information about system events should be recorded in auditing systems. The time and user involved in each system login can be logged. Monitoring of system activities can detect unusual activities.
- Virus checkers can be used to search the presence of known abnormalities.
- The existence of a long-running process in a listing of currently executing processes may indicate suspicious activity.
- The current state of the system can be checked against a previous state.

11.6 Correction

If a system has been penetrated, it is necessary to take corrective action.

- Periodic backup should be performed to rollback the system to a previous state.
- If backup does not exist or its integrity is unknown, entire system may be reloaded.
- It may be necessary to change all security information. All users may be required to change their passwords.
- The cause that allowed the system to be penetrated should be fixed. It may involve deactivating a service, installing bug fix or modifying the system configuration.

11.7 Identification

The source of attack should be identified to discourage intruders. It is the most difficult security task.

- Audit records provide useful identification information. But the information in audit logs may have been tamped with by the intruder
- System accessed through modems can keep track of source of incoming calls using user-id.
- System accessed through a network can record the address of the connecting computer. Attacks through a series of computers must be traced to their origin.
- All services can be configured to require user authentication. A mail server can refuse mail services to any unauthorized client. If that server is used to relay an email virus, the authentication information may be useful in identifying its source.

11.8 Program Threats

Often one user writes a program to be used by another user. In these cases, misuse can occur and lead to unexpected behavior. The two most common methods by which such behavior may occur are Trojan horses and trap doors.

11.8.1 Trojan Horse

One of the most serious threats to computer security is Trojan horse attack. A Trojan horse is a malicious, security-breaking program that is disguised as something harmless such as a screen saver or game. A famous Trojan horse was the so-called "Love Bug".

When programs are written by users to be executed by other users, the user can misuse the rights of the program if they execute the program in a domain that provides the access rights of the executing user.

One can end such program by ending the interactive session with a sequence of key pressing such as control-alt-delete combination in Windows 95/98/NT systems.

11.8.2 Trap Doors

When a programmer leaves a hole in the software, only he is capable of gaining access to it. A trap door is basically a program where the programmer might illegally or legally write the code to avoid normal security procedures for specific user.

Trap doors are difficult problems to detect because they are in the source code where software systems can contain millions of lines of code.

11.9 System Threats

Most operating systems provide a means for processing to offspring other processes. With an environment like that it becomes easy for operating-system resources and user file to be misused. Worms and viruses are the most common methods for achieving misuse.

11.9.1 Worms

A worm is a process that uses the spawn (or offspring) mechanism to strike system performance. The way it works is the worm spawns copies of it by using system resources. It may even lock out system for all other processes. On a network system, worms are usually strong, since they multiply and eventually shut down the entire network.

11.9.2 Viruses

Another common computer attack is called a virus. Viruses are designed to spread into programs that can cause disorder in a system similarly like worms. Viruses can also modify or destroy files. A worm is structured in a complete form like separate self-program, while a virus is a portion of embedded code that begins in a legitimate program. Viruses can attack a user seriously and can become a big problem especially to microcomputer users.

The most common way of receiving a virus is the downloading programs, downloading files from Internet or exchanging floppy disks that contain a virus infection. Viruses can be avoided by using antivirus software that automatically starts every time the PC is turned on.

11.10 Threat Monitoring

Threat monitoring is a management technique that can improve a security system. This system can easily check any suspicious activity to violate security. A good example of threat monitoring is when a user is attempting to log in. The system may count the number of incorrect passwords given when trying to log in. After a few attempts of incorrect password input, a signal is sent to warn that an intruder might be trying to guess the password.

Another common technique is an audit log. An audit log records information such as time, user name and type of accesses to an object. If a sign of security violation occurs, a collection of data is recorded to determine how and when the violation occurred.

A scanning method can be used to scan the computers to check for security holes. A scan looks for the following aspects of a system:

- Short or easy-to-guess passwords

- Unauthorized programs in system directories.
- Unexpected long-running process
- Improper directory protections, on both user and system directories
- Improper protections on system data files, such as password file, device drivers, or even the operating-system kernel itself
- Dangerous entries in the program search path (i.e. Trojan horse)
- Changes to system programs detected with checksum values

When problems are found by the security scan, they be automatically fixed or be directly reported to the managers of the system.

Internet is a main source of security problems as it connects millions of computers. One solution to protection and security through the Internet is a **firewall**. A firewall is a computer or router that sits between trusted and un-trusted. It limits network access between the two security domains, and monitors logs and connections. The following figure shows how network security of a firewall may work for a company. **DMZ stands for demilitarizes zone.**

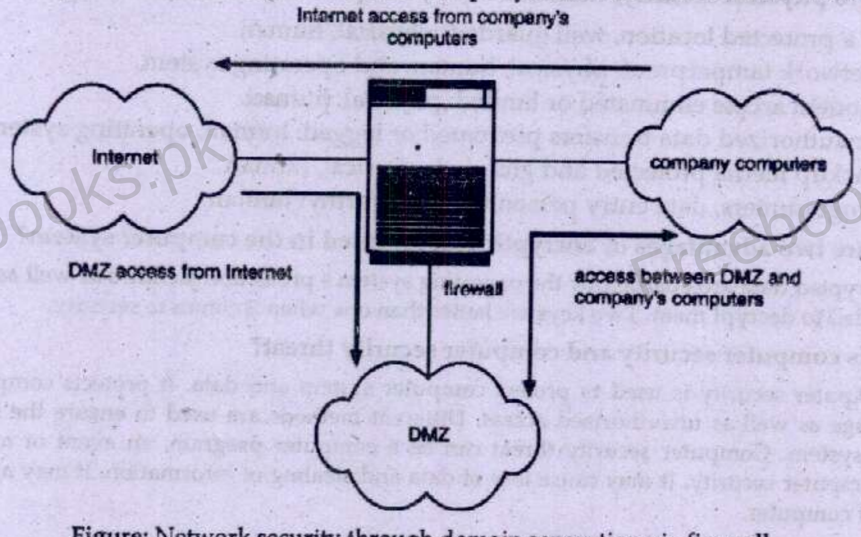

Internet access from company's computers
Internet
company computers
DMZ access from Internet
firewall
access between DMZ and company's computers
DMZ

Figure: Network security through domain separation via firewall.

11.11 System Security

No single security solution can provide total system security. A proper balance of security mechanisms must be achieved. Each security mechanism provides specific security functions and should be designed to only provide those separate functions. It should rely on other mechanisms for support and for required security services.

In a secure system, the entire set of mechanisms complement each other so that they collectively provide a complete security package. Systems that fail to achieve this balance will be vulnerable. A secure operating system is an important and necessary piece to the total system security.

A highly secure operating system would be insufficient without an application-specific security built on it. Certain problems are actually better addressed by security implemented above the operating system. One such example is an electronic commerce system that requires a digital signature on each transaction. A cryptographic mechanism in the transaction system, protected by secure operating system features, might offer the best system security solution.

Review Questions

1. Is there a method to detect that password has become known to others?

Whenever a user logs in, the system prints the last time that user was logged on the system.

2. List three effects that indicate the security threats.

The computer runs slower than usual. The programs are loaded very slowly. The application software and disk drives cannot be accessed.

3. The list of all passwords is kept within the operating system. Thus, if a user manages to read this list, password protection is no longer provided. Suggest a scheme that will avoid this problem.

Encrypt the passwords internally so that they can only be accessed in coded form. The only person with access or knowledge of decoding should be the system operator.

4. List the security concerns for a computer system for a bank. State whether the concern relates to physical security, human security or operating system security.

- In a protected location, well guarded: physical, human.
- Network tamperproof: physical, human, and operating system.
- Modem access eliminated or limited: physical, human.
- Unauthorized data transfers prevented or logged: human, operating system.
- Backup media protected and guarded: physical, human.
- Programmers, data entry personnel, trustworthy: human.

5. What are two advantages of encrypting data stored in the computer system?

Encrypted data are guarded by the operating system's protection facilities as well as a password that is needed to decrypt them. Two keys are better than one when it comes to security.

6. What is computer security and computer security threat?

Computer security is used to protect computer system and data. It protects computer system from damage as well as unauthorized access. Different methods are used to ensure the security of a computer system. Computer security threat can be a computer program, an event or a person that violates computer security. It may cause loss of data and stealing of information. It may also affect the working of computer.

7. Define virus.

Computer virus is a program that may disturb the normal working of computer without the knowledge or permission of the user. It is the most common and well known computer security threat. Virus attaches itself to the files in flash drives, hard disks and email attachments. A file containing a virus is called infected file. The virus is activated when the infected file is executed.

8. Write different problems caused by virus.

Virus can cause many problems. It may:

- Damage the files stored on the computer
- Erase data stored on the disk
- Affect the normal working of the computer
- Display unusual messages

9. What is the difference between virus and worm?

Worm can spread copies of itself from one computer to another without being activated by users. A virus attaches itself to a program or file to spread from one computer to another. Virus cannot spread without a human action such as running an infected program.

12 CASE STUDY: LINUX

Chapter Overview

12.1 Introduction
12.2 Brief History of Linux
12.3 Features of Linux
12.4 Who is using Linux?
12.5 Linux Distributions
12.6 Linux Architecture: System Structure
12.7 Process Scheduler Subsystem
    12.7.1 Goals
    12.7.2 External Interface
    12.7.3 Subsystem Description
    12.7.4 Data Structures
    12.7.5 Subsystem Structure
    12.7.6 Subsystem Dependencies
12.8 Memory Manager Subsystem
    12.8.1 Goals
    12.8.2 External Interface
    12.8.3 Subsystem Description
    12.8.4 Data Structures
    12.8.5 Subsystem Structure
    12.8.6 Subsystem Dependencies
12.9 Virtual File System
    12.9.1 Goals
    12.9.2 External Interface
    12.9.3 Subsystem Description
    12.9.4 Data Structures
    12.9.5 Subsystem Structure
12.9.6 Subsystem Dependencies
12.10 Inter-Process Communication
    12.10.1 Goals
    12.10.2 External Interface
    12.10.3 Subsystem Description
    12.10.4 Data Structures
    12.10.5 Subsystem Structure
    12.10.6 Subsystem Dependencies
12.11 Network Interface
    12.11.1 Goals
    12.11.2 External Interface
    12.11.3 Subsystem Description
    12.11.4 Data Structures
    12.11.5 Subsystem Structure
    12.11.6 Subsystem Dependencies
Review Questions

12.1 Linux

Linux is another UNIX-like system that has gained popularity in recent years. It has been designed to run as many standard UNIX applications as possible. It has much in common with existing UNIX implementations. Linux is a rapidly evolving operating system.

12.2 Brief History of Linux

MS DOS was popularly used in the early 1990's by many personal computers. **Apple Macs** were better but the price was very high. **Unix** itself was far more expensive. An operating system **MINIX** was written by Andrew S. Tanenbaum who was a Dutch professor. It was designed to run on Intel 8086 microprocessors. It was not outstanding but its source code was available. **Linux Torvalds** re-wrote MINIX and created **Linux**. The Linux operating system has also been adapted for use in Alpha, Mac, PowerPC and even palmtops.

12.3 Features of Linux

Some features of Linux are as follows:

1. Multi-platform

Linux is a cross-platform operating system. It runs on many computer models.

2. Free

Linux is free. The user can purchase Linux from a vendor who bundles Linux with special documentation or applications or provides technical support.

3. Open Source

Linux and many Linux applications are distributed in source form. It allows others to modify or improve them. That is why Linux is constantly improved and updated rapidly. It will likely be the first operating system to support Intel's forthcoming Merced 64-bit CPU.

4. Hardware

Linux can run on almost any hardware such as 386, 486, Pentium MMX, Pentium II, Sparc, Dec Alpha or even Motorola 68000 series.

5. Multi-Tasking

Linux is a multi-tasking system. It means that a single user can run multiple programs at the same time. Each task is called a process. It means that a user can give the system a command to be run in the background while doing more important work in the foreground. This way a user does not have to wait for a process to finish in order to start another one.

6. Multi-User

Multiuser means that hundreds of people can use the computer at the same time over a network, Internet or laptops/computers or terminals connected to serial ports of computers.

7. Multiple Virtual Consoles

It provides multiple virtual terminals on server. It can be done by pressing a key combination. It enables different users to login at same time on same machine.

8. TCP/IP Networking

Linux is one of the best operating systems in terms of networking. **TCP/IP** is a set of protocols that links millions of university and business computers into a worldwide network known as Internet. With an Ethernet connection, the user can access the Internet or LAN from Linux system. The user can use **SLIP (Serial Line Internet Protocol)** to access Internet over phone lines with a modem. Linux makes it easy to construct firewalls to protect system.

9. High Level Security

Linux provides a very high level of security by using user authentication. It also stores passwords in an encrypted form. The password once encrypted cannot be decrypted. Linux also includes file system security that enhances the existing security.

10. GNU Software Support

Linux supports a wide range of free software written by **GNU Project**. It includes utilities such as **GNU C** and **C++** compiler, **gawk**, **groff** etc. Many essential system utilities used by Linux are GNU software.

11. Virtual Memory and Shared Libraries

Linux can use a portion of hard drive as virtual memory to expand total amount of available RAM. It also implements shared libraries. It allows programs that use standard subroutines to find the code for these subroutines in libraries at runtime. It saves a large amount of space as each application doesn't store its own copy of common routines.

12. Stable Operating System

Linux is a complete operating system that is stable. It means that the malfunctioning of an application is not likely to bring the system down.

13. Reliable

Linux servers are very reliable. They remain up for hundreds of days compared to the regular reboots seen in the case of other operating systems.

14. Web Server

Linux can be used to run a web server such as **Apache** to serve application protocols such as **HTTP** or **FTP**.

15. Multiple Processor Support

Linux supports multiple processors. Many companies have hardware that is in danger of becoming obsolete. The capability of adding processors means that these companies will be able to continue using the existing hardware thereby safeguarding their investment.

12.4 Who is using Linux?

For years, only people who appreciated Linux were the hackers and the engineers who helped develop it. Linux had a relatively small user base for some basic reasons. American public wanted and American government allowed for a single OS standard to dominate the computer industry. Microsoft became a near monopoly because of that dominance.

In the last two years, Linux suddenly emerged as a potential competitor to Windows. Some Microsoft competitors such as Hewlett Packard and IBM had funded to commercialize Linux. Corporate America is likely to move to Linux if popular software applications are redesigned to run on Linux.

12.5 Linux Distributions

Linux can be freely redistributed. The user can obtain it in a variety of ways. Various individuals and organizations package Linux, often combining it with free or proprietary applications. Such a package that includes all the software needed to install and run Linux is called a **Linux distribution**.

Caldera, Red Hat, Slackware and **SuSE** are packaged by commercial companies that get profit by selling Linux-related products and services. The user can download these distributions from the respective companies' web sites or make additional copies of a Linux distribution. **Debian GNU/Linux** is a product of volunteer effort conducted under auspices of Software **In The Public Interest, Inc.**.

12.6 Linux Architecture: System Structure

Linux Kernel participates as one layer in overall system. Linux is composed of five major subsystems within the kernel layer:

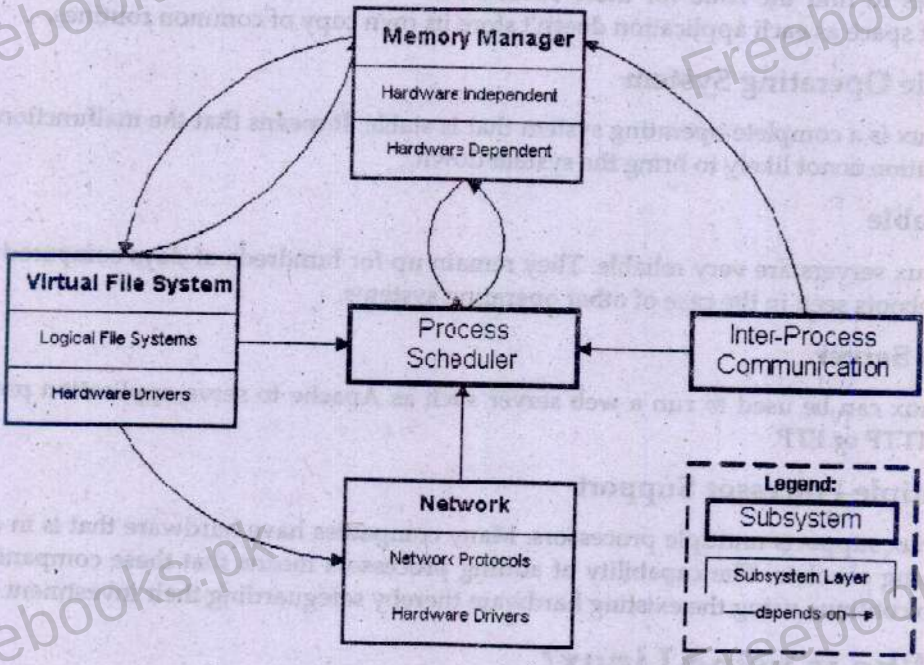

Memory Manager
Hardware Independent
Hardware Dependent
Virtual File System
Logical File Systems
Hardware Drivers
Process Scheduler
Inter-Process Communication
Network
Network Protocols
Hardware Drivers
Legend:
Subsystem
Subsystem Layer
depends on

Figure: Conceptual Decomposition

- The process scheduler (sched)
- The memory manager (mm)
- The virtual file system (vfs)
- The network interface (net)
- The inter-process communication (ipc).

12.7 Process Scheduler Subsystem

12.7.1 Goals

Process scheduling is the heart of Linux operating system. The process scheduler has the following responsibilities:

- Allow processes to create new copies of themselves
- Determine which process will have access to the CPU and effect the transfer between running processes
- Receive interrupts and route them to the appropriate kernel subsystem
- Send signals to user processes
- Manage the timer hardware
- Clean up process resources when a processes finishes executing

The process scheduler also provides support for dynamically loaded modules. These modules represent kernel functionality that can be loaded after kernel has started executing. The loadable module functionality is used by virtual file system and network interface.

12.7.2 External Interface

The process scheduler provides two interfaces. First, it provides a limited system call interface that user processes may call. Secondly, it provides a rich interface to the rest of the kernel system.

Processes can only create other processes by copying the existing process. At boot time, Linux system has only one running process: init. This process then spawns others that can also spawn off copies of themselves through fork () system call. The fork () call generates a new child process that is a copy of its parent. Upon termination, a user process (implicitly or explicitly) calls the _exit () system call.

Several routines are provided to handle loadable modules. A create_module () system call will allocate enough memory to load a module. The call will initialize the module structure, described below, with the name, size, starting address, and initial status for the allocated module. The init_module () system call loads the module from disk and activates it. Finally, delete_module () unloads a running module.

Timer management can be done through the setitimer() and getitimer() routines. The former sets a timer while the latter gets a timer's value.

Among the most important signal functions is signal(). This routine allows a user process to associate a function handler with a particular signal.

12.7.3 Subsystem Description

The process scheduler subsystem is primarily responsible for the loading, execution, and proper termination of user processes. The scheduling algorithm is called at two different points during the execution of a user process. First, there are system calls that call the scheduler directly, such as sleep(). Second, after every system call, and after every slow system interrupt (described in a moment), the schedule algorithm is called.

Signals can be considered an IPC mechanism, thus are discussed in the inter-process communication section.

Interrupts allow hardware to communicate with the operating system. Linux distinguishes between slow and fast interrupts. A slow interrupt is a typical interrupt. Other interrupts are legal while they are being processed, and once processing has completed on a slow interrupt, Linux conducts business as usual, such as calling the scheduling algorithm. A timer interrupt is exemplary of a slow interrupt. A fast interrupt is one that is used for much less complex tasks, such as processing keyboard input. Other interrupts are disabled as they are being processed, unless explicitly enabled by the fast interrupt handler.

The Linux OS uses a timer interrupt to fire off once every 10ms. Thus, according to our scheduler description given above, task rescheduling should occur at lease once every 10ms.

12.7.4 Data Structures

The structure task_struct represents a Linux task. There is a field that represents the process state; this may have the following values:

- Running
- Returning from system call
- Processing an interrupt routine
- Processing a system call
- Ready
- Waiting

In addition, there is a field that indicates the processes priority, and field, which holds the number of clock ticks (10ms intervals), which the process can continue executing without, forced rescheduling. There is also a field that holds the error number of the last faulting system call.

In order to keep track of all executing processes, a doubly linked list is maintained, (through two fields that point to task_struct). Since every process is related to some other process, there are fields which describe a processes: original parent, parent, youngest child, younger sibling, and finally older sibling. There is a nested structure, mm_struct, which contains a process's memory management information, (such as start and end address of the code segment).

Process ID information is also kept within the task_struct. The process and group id are stored. An array of group id's is provided so that a process can be associated with more than one group. File specific process data is located in a fs_struct substructure. This will hold a pointer to the inode corresponding to a processors root directory, and it's current working directory.

All files opened by a process will be kept track of through a files_struct substructure of the task_struct. Finally, there are fields that hold timing information; for example, the amount of time the process has spent in user mode.

All executing processes have an entry in the process table. The process table is implemented as an array of pointers to task structures. The first entry in the process table is the special init process, which is the first process executed by the Linux system.

Finally, a module structure is implemented to represent the loaded modules. This structure contains fields that are used to implement a list of module structure: a field which points to the modules symbol table, and another field that holds the name of the module. The module size (in pages), and a pointer to the starting memory for the module are also fields within the module structure.

12.7.5 Subsystem Structure

The below figure shows the Process Scheduler subsystem. It is used to represent, collectively, process scheduling and management (i.e. loading and unloading), as well as timer management and module management functionality.

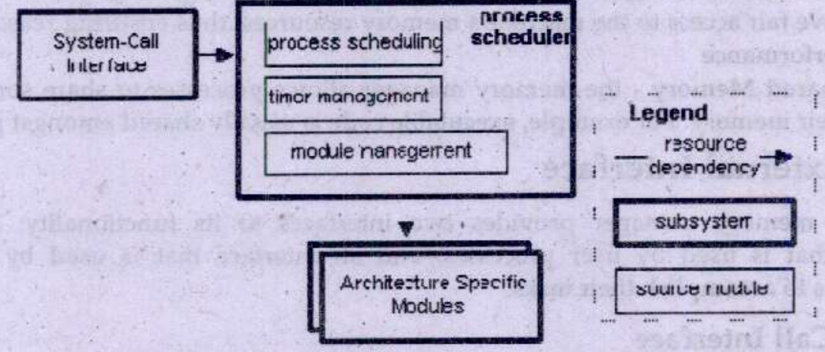

System-Call Interface
process scheduling
process scheduler
timer management
module management
Legend
resource dependency
subsystem
source module
Architecture Specific Modules

Figure: Process Scheduler Structure

12.7.6 Subsystem Dependencies

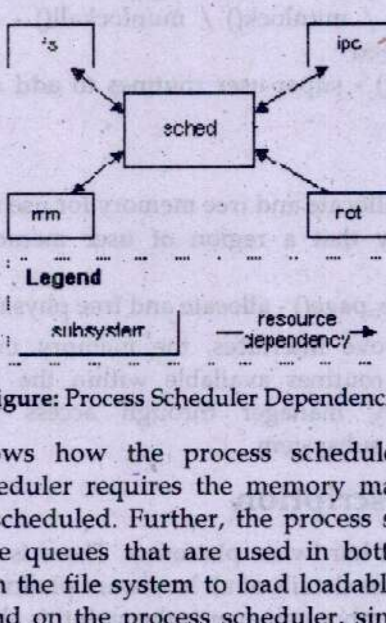

fs
ipc
sched
mm
net
Legend
subsystem
resource dependency

Figure: Process Scheduler Dependencies

The above figure shows how the process scheduler depends on other kernel subsystems. The process scheduler requires the memory manager to set up the memory mapping when a process is scheduled. Further, the process scheduler depends on the IPC subsystem for the semaphore queues that are used in bottom-half-handling Finally, the process scheduler depends on the file system to load loadable modules from the persistent device. All subsystems depend on the process scheduler, since they need to suspend user processes while hardware operations complete.

12.8 Memory Manager Subsystem

12.8.1 Goals

The memory manager provides the following capabilities to its clients:

- **Large address space** - user programs can reference more memory than physically exists

- Protection - the memory for a process is private and cannot be read or modified by another process; also, the memory manager prevents processes from overwriting code and read-only-data.
- **Memory Mapping** - clients can map a file into an area of virtual memory and access the file as memory
- **Fair Access to Physical Memory** - the memory manager ensures that processes all have fair access to the machine's memory resources, thus ensuring reasonable system performance
- **Shared Memory** - the memory manager allows processes to share some portion of their memory. For example, executable code is usually shared amongst processes.

12.8.2 External Interface

The memory manager provides two interfaces to its functionality: a system-call interface that is used by user processes, and an interface that is used by other kernel subsystems to accomplish their tasks.

System Call Interface

- malloc() / free() - allocate or free a region of memory for the process's use
- mmap() / munmap() / msync() / mremap() - map files into virtual memory regions
- mprotect - change the protection on a region of virtual memory
- mlock() / mlockall() / munlock() / munlockall() - super-user routines to prevent memory being swapped
- swapon() / swapoff() - super-user routines to add and remove swap files for the system

Intra-Kernel Interface

- kmalloc() / kfree() - allocate and free memory for use by the kernel's data structures
- verify_area() - verify that a region of user memory is mapped with required permissions
- get_free_page() / free_page() - allocate and free physical memory pages

In addition to the above interfaces, the memory manager makes all of its data structures and most of its routines available within the kernel. Many kernel modules interface with the memory manager through access to the data structures and implementation details of the subsystem.

12.8.3 Subsystem Description

Linux supports several hardware platforms. There is a platform-specific part of the memory manager that abstracts details of all hardware platforms into one common interface. All access to the hardware memory manager is through this abstract interface.

The memory manager uses hardware memory manager to map virtual addresses to physical memory addresses. When a user process accesses a memory location, the hardware memory manager translates virtual memory address to a physical address and then uses the physical address to perform the access. Because of this mapping, user processes are not aware of what physical address is associated with a particular virtual memory address. It allows memory manager subsystem to move process's memory around in physical memory. The mapping also permits two user processes to share physical memory if regions of their virtual memory address space map to the same physical address space.

The memory manager swaps process memory out to a paging file when it is not in use. It allows the system to execute processes that use more physical memory than available memory. The memory manager contains a daemon (kswapd). Linux uses the term daemon to refer to kernel threads; a daemon is scheduled by the process scheduler in the same way that user processes are, but daemons can directly access kernel data structures. The concept of a daemon is closer to a thread than a process.

The kswapd daemon periodically checks to see if there are any physical memory pages that have not been referenced recently. These pages are evicted from physical memory. If necessary, they are stored on disk. The memory manager subsystem takes special care to minimize the amount of disk activity that is required. The memory manager avoids writing pages to disk if they could be retrieved another way.

The hardware memory manager detects when a user process accesses a memory address that is not currently mapped to a physical memory location. The hardware memory manager notifies Linux kernel of this page fault. It is up to memory manager subsystem to resolve the fault. There are two possibilities: either the page is currently swapped out to disk, and must be swapped back in, or the user process is making an invalid reference to a memory address outside of its mapped memory. The hardware memory manager also detects invalid references to memory addresses such as writing to executable code or executing data. These references also result in page faults that are reported to the memory manager subsystem. If the memory manager detects an invalid memory access, it notifies the user process with a signal; if the process doesn't handle this signal, it is terminated.

12.8.4 Data Structures

The following data structures are relevant:

vm_area	The memory manager stores a data structure with each process that records what regions of virtual memory are mapped to which physical pages. It also stores a set of function pointers that allow it to perform actions on a particular region of the process's virtual memory.
mem_map	The memory manager maintains a data structure for each page of physical memory on system. This data structure contains flags that indicate the status of the page. All page data structures are available in a vector that is initialized at kernel boot time. As page status changes, the attributes in data structure are updated.
free_area	The free_area vector is used to store unallocated physical memory pages; pages are removed from the free_area when allocated, and returned when freed.

12.8.5 Subsystem Structure

The memory manager subsystem is composed of several source code modules; these can be decomposed by areas of responsibility into the following groups (shown in Figure):

- **System Call Interface:** This group of modules is responsible to present the services of memory manager to user processes through a well-defined interface.
- **Memory-Mapped Files (MMAP):** This group of modules is responsible for supported memory-mapped file I/O.

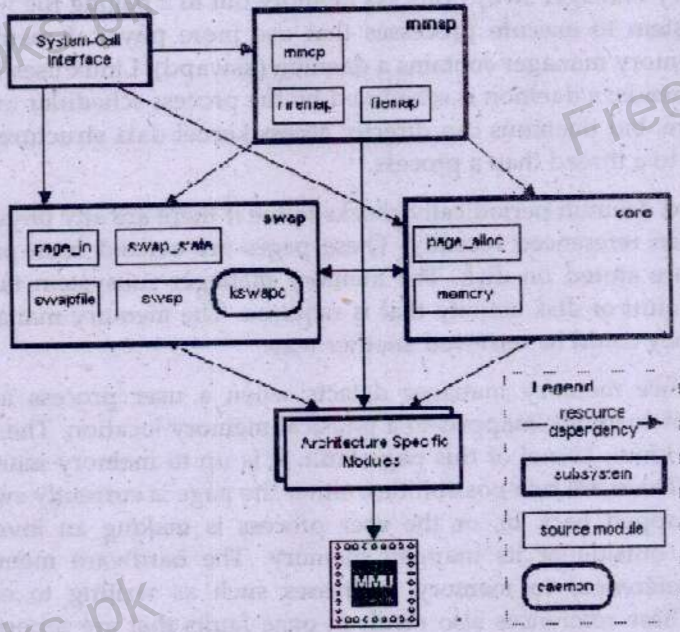

System-Call interface
mmap
mmap
mremap
filemap
swap
page_in
swap_state
swapfile
swap
kswapd
core
page_alloc
memory
Legend
resource dependency
subsystem
source module
daemon
Architecture Specific Modules
MMU

Figure: Memory Manager Structure

- **Swap file Access (Swap):** This group of modules controls memory swapping. These modules initiate page-in and page-out operations.
- **Core Memory Manager (Core):** These modules are responsible for the core memory manager functionality that is used by other kernel subsystems.
- **Architecture Specific Modules:** These modules provide a common interface to all supported hardware platforms. These modules execute commands to change the hardware MMU's virtual memory map, and provide a common means of notifying the rest of the memory-manager subsystem when a page fault occurs.

12.8.6 Subsystem Dependencies

The memory manager is used directly (via data structures and implementation functions) by each of sched, fs, ipc, and net.

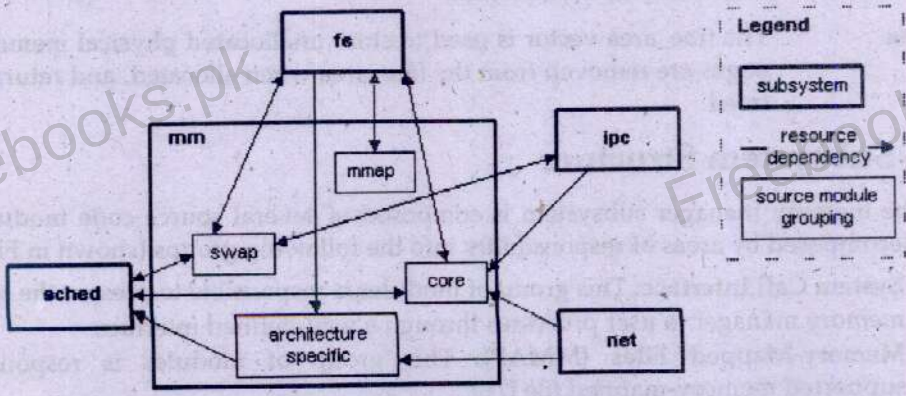

fs
mm
mmap
swap
core
architecture specific
sched
ipc
net
Legend
subsystem
resource dependency
source module grouping

Figure: Memory Manager Dependencies

12.9 Virtual File System

12.9.1 Goals

Linux is designed to support many different physical devices. Even for one specific type of device, such as hard drives, there are many interface differences between different hardware vendors. Linux supports a number of logical file systems. It can inter-operate easily with other operating systems. The Linux file system supports the following goals:

- Multiple hardware devices - provide access to many different hardware devices
- Multiple logical file systems - support many different logical file systems
- Multiple executable formats - support several different executable file formats (like a.out, ELF, java)
- Homogeneity - present a common interface to all of the logical file systems and all hardware devices
- Performance - provide high-speed access to files
- Safety - do not lose or corrupt data
- Security - restrict user access to access files; restrict user total file size with quotas

12.9.2 External Interface

The file system provides two levels of interface: a system-call interface that is available to user processes, and an internal interface that is used by other kernel subsystems. The system-call interface deals with files and directories. Operations on files include the usual open/close/read/write/seek/tell that are provided by POSIX compliant systems; different types of operations on directories include readdir/creat/unlink/chmod/stat as usual for POSIX systems.

The interface that the file subsystem supports for other kernel subsystems is much richer. The file subsystem exposes data structures and implementation function for direct manipulation by other kernel subsystems. In particular, two interfaces are exposed to the rest of the kernel -- inodes and files. Other implementation details of the file subsystem are also used by other kernel subsystems, but this use is less common.

Inode Interface

- create(): create a file in a directory
- lookup(): find a file by name within a directory
- link() / symlink() / unlink() / readlink() / follow_link(): manage file system links
- mkdir() / rmdir(): create or remove sub-directories
- mknod(): create a directory, special file, or regular file
- readpage() / writepage(): read or write a page of physical memory to a backing store
- truncate(): set the length of a file to zero
- permission(): check to see if a user process has permission to execute an operation
- smap(): map a logical file block to a physical device sector
- bmap(): map a logical file block to a physical device block
- rename(): rename a file or directory

In addition to the methods you can call with an inode, the namei() function is provided to allow other kernel subsystems to find the inode associated with a file or directory.

File Interface

- open() / release(): open or close the file
- read() / write(): read or write to the file
- select(): wait until the file is in a particular state (readable or writeable)
- lseek(): if supported, move to a particular offset in the file
- mmap(): map a region of the file into the virtual memory of a user process
- fsync() / fasync(): synchronize any memory buffers with the physical device
- readdir: read the files that are pointed to by a directory file
- ioctl: set file attributes
- check_media_change: check to see if a removable media has been removed (such as a floppy)
- revalidate: verify that all cached information is valid

12.9.3 Subsystem Description

The file subsystem needs to support many different logical file systems and many different hardware devices. It does this by having two conceptual layers that are easily extended. The device driver layer represents all physical devices with a common interface. The **virtual file system layer (VFS)** represents all logical file systems with a common interface. The conceptual architecture of Linux kernel shows how this decomposition is conceptually arranged.

Device Drivers

The device driver layer is responsible for presenting a common interface to all physical devices. Linux kernel has three types of device driver: **character, block** and **network**. The two types relevant to file subsystem are character and block devices. Character devices must be accessed sequentially such as tape drives, modems, and mice. Block devices can be accessed in any order but can only be read and written to in multiples of the block size.

Each device can be accessed as though it was a file in file system. This file is referred to as a device special file. Most of kernel deals with devices via file interface, it is easy to add a new device driver by implementing hardware-specific code to support this file interface.

Linux kernel uses a buffer cache to improve performance when accessing block devices. All access to block devices occurs through a buffer cache subsystem. The buffer cache greatly increases system performance by minimizing reads and writes to hardware devices. Each hardware device has a request queue; when the buffer cache cannot fulfill a request from in-memory buffers, it adds a request to the device's request queue and sleeps until this request has been satisfied. The buffer cache uses a separate kernel thread, kflushd, to write buffer pages out to the devices and remove them from the cache.

When a device driver needs to satisfy a request, it begins by initiating the operation with hardware device manipulating the device's control and status registers (CSR's). There are three general mechanisms for moving data from the main computer to peripheral device: polling, direct memory access (DMA) and interrupts.

When a hardware device wants to report a change in condition (mouse button pushed, key pressed) or to report the completion of an operation, it sends an interrupt to the CPU. If interrupts are enabled, CPU stops executing current instruction and begins executing Linux kernel's interrupt handling code. The kernel finds appropriate interrupt handler to invoke. While an interrupt is being handled, CPU executes in a special context; other interrupts may be delayed until the interrupt is handled. Because of this restriction, interrupt handlers need to be quite efficient so that other interrupts are not lost. Sometimes an interrupt handler cannot complete all required work within the time constraints. In this case, the interrupt handler schedules the work in a bottom-half handler. A bottom-half handler is code that is executed by scheduler the next time a system call is completed.

Logical File Systems

It is possible to access physical devices through device special file. It is more common to access block devices through a logical file system. A logical file system can be mounted at a mount point in virtual file system. It means that the associated block device contains files and structure information that allow logical file system to access the device. At any one time, a physical device can only support one logical file system. However, the device can be reformatted to support a different logical file system.

When a file system is mounted as a subdirectory, all directories and files available on the device are made visible as subdirectories of mount point. Users of virtual file system do not need to be aware what logical file system is implementing which parts of directory tree etc. This abstraction provides a great deal of flexibility in the choice of physical devices and logical file systems. It is one of the essential factors in success of Linux operating system.

Linux uses the concept of inodes to support virtual file system. It uses an inode to represent a file on a block device. The inode is virtual in the sense that it contains operations that are implemented differently depending on logical system and physical system. The inode interface makes all files appear the same to other Linux subsystems. It is used as a storage location for all of the information related to an open file on disk. It stores associated buffers, total length of file in blocks and the mapping between file offsets and device blocks.

Modules

Most of the functionality of virtual file system is available in the form of dynamically loaded modules. This dynamic configuration allows Linux users to compile a kernel that is as small as possible, while still allowing it to load required device driver and file system modules if necessary during a single session. For example, a Linux system might optionally have a printer attached to its parallel port. If printer driver were always linked in to kernel, then memory would be wasted when the printer isn't available. By making the printer driver be a loadable module, Linux allows the user to load the driver if the hardware is available.

12.9.4 Data Structures

The following data structures are architecturally relevant to the file subsystem:

super_block	Each logical file system has associated superblock used to represent it to rest of Linux kernel. It contains information about the entire mounted file system.
Inode	An inode is an in-memory data structure that represents all of the information that the kernel needs to know about a file on disk. It stores all of the information that the kernel needs to associate with a single file. Accounting, buffering, and memory mapping information are all stored in the inode.
File	The file structure represents a file that is opened by a particular process. All open files are stored in a doubly-linked list. The file descriptor used in POSIX style routines (open, read, write) is the index of a particular open file in this linked list.

12.9.5 Subsystem Structure

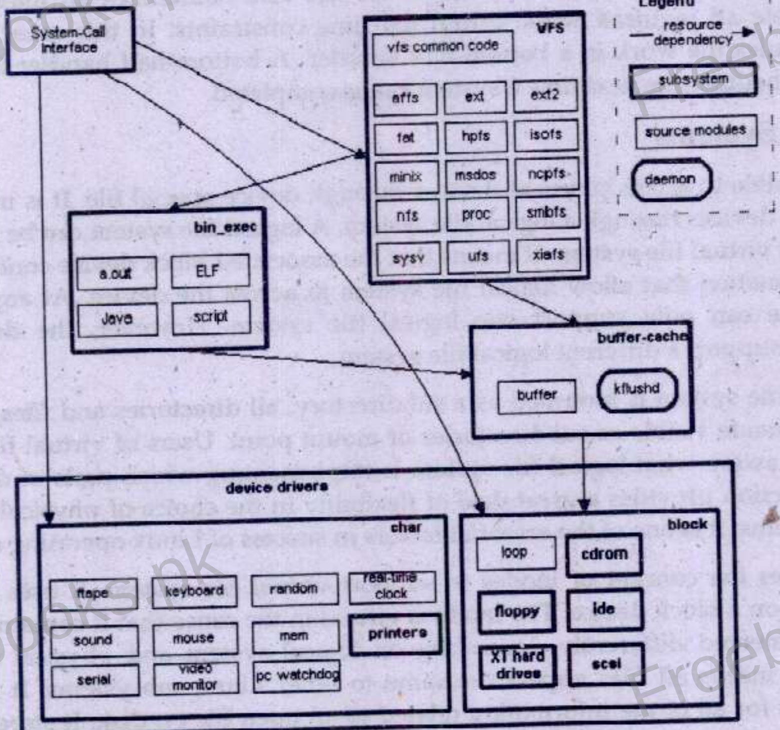

System-Call Interface
VFS
vfs common code
affs
ext
ext2
fat
hpfs
isofs
minix
msdos
ncpfs
nfs
proc
smbfs
sysv
ufs
xiafs
bin_exec
a.out
ELF
Java
script
buffer-cache
buffer
kflushd
device drivers
char
flope
keyboard
random
real-time clock
sound
mouse
mem
printers
serial
video monitor
pc watchdog
block
loop
cdrom
floppy
ide
XT hard drives
scsi
Legend
resource dependency
subsystem
source modules
daemon

Figure: File Subsystem Structure

12.9.6 Subsystem Dependencies

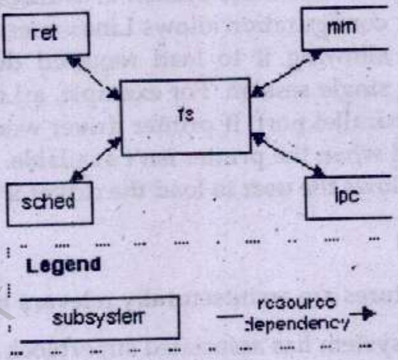

net
mm
fs
sched
ipc
Legend
subsystem
resource dependency

Figure: File Subsystem Dependencies

The figure shows how the file system is dependent on other kernel subsystems. The file system depends on all other kernel subsystems and all other kernel subsystems depend on the file subsystem. In particular, the network subsystem depends on the file system because network sockets are presented to user processes as file descriptors. The memory manager depends on file system to support swapping. The IPC subsystem depends on file system to implement pipes and FIFO's. The process scheduler depends on file system to load modules.

The file system uses the network interface to support NFS. It uses memory manager to implement buffer cache and for a ramdisk device. It uses IPC subsystem to help support modules and it uses process scheduler to put user processes to sleep while hardware requests are completed.

12.10 Inter-Process Communication

12.10.1 Goals

Linux IPC mechanism is provided so that concurrently executing processes have a means to share resources, synchronize and exchange data with one another. Linux implements all forms of IPC between processes executing on the same system through shared resources, kernel data structures, and wait queues.

Linux provides the following forms of IPC:

Signals	It is perhaps the oldest form of Unix IPC, signals are asynchronous messages sent to a process.
Wait queues	It provides a mechanism to put processes to sleep while they are waiting for an operation to complete.
File locks	It provides a mechanism to allow processes to declare either regions of a file, or the entire file itself, as read-only to all processes except the one which holds the file lock.
Pipes and Named Pipes	It allows connection-oriented, bi-directional data transfer between two processes either by explicitly setting up the pipe connection, or communicating through a named pipe residing in the file-system.
Semaphores	It is an implementation of a classical semaphore model. The model also allows for the creation of arrays of semaphores.
Message queues	It is a connectionless data-transfer model. A message is a sequence of bytes, with an associated type. Messages are written to message queues, and messages can be obtained by reading from the message queue, possibly restricting which messages are read in by type.
Shared memory	It is a mechanism by which several processes have access to the same region of physical memory.
Unix Domain sockets	It is another connection-oriented data-transfer mechanism that provides the same communication model as the INET sockets, discussed in the next section.

12.10.2 External Interface

A signal is a notification sent to a process by kernel or another process. Signals are sent with the send_sig() function. The signal number is provided as a parameter, as well as the destination process. Processes may register to handle signals by using the signal() function.

File locks are supported directly by the Linux file system. To lock an entire file, the open() system call can be used, or the sys_fcntl() system-call can be used. Locking areas within a file is done through the sys_fcntl() system call.

Pipes are created by using the pipe() system call. The file-systems read() and write() calls are then used to transfer data on the pipe. Named pipes are opened using the open() system-call. The System V IPC mechanisms have a common interface, which is the ipc() system call. The various IPC operations are specified using parameters to the system call.

The Unix domain socket functionality is also encapsulated by a single system call, socketcall(). Each of the system-calls mentioned above are well documented, and the reader is encouraged to consult the corresponding man-page.

The IPC subsystem exposes wait calls to other kernel subsystems. Since wait queues are not used by user processes, they do not have a system-call interface. Wait queues are used in implementing semaphores, pipes, and bottom-half handlers (see section 3.3.3). The procedure add_wait_queue() inserts a task into a wait queue. The procedure remove_wait_queue() removes a task from the wait queue.

12.10.3 Subsystem Description

Signals are used to notify a process of an event. A signal has the effect of altering the state of recipient process, depending on the semantics of particular signal. Kernel can send signals to any executing process. A user process may only send a signal to a process or process group if it possesses associated PID or GID. Signals are not handled immediately for dormant processes. Rather, before the scheduler sets a process running in user mode again, it checks if a signal was sent to process. If so, then the scheduler calls the do_signal() function, which handles the signal appropriately.

Wait queues are simply linked lists of pointers to task structures that correspond to processes that are waiting for a kernel event such as conclusion of a DMA transfer. A process can enter itself on the wait queue by either calling sleep_on() or interruptable_sleep_on() functions. The functions wake_up() and wake_up_interruptable() remove the process from the wait queue. Interrupt routines also use wait-queues to avoid race conditions.

Linux allows user process to prevent other processes to access a file. This exclusion can be based on a whole file or a region of a file. File-locks are used to implement this exclusion. The file-system implementation contains appropriate data fields in its data structures to allow kernel to determine if a lock has been placed on a file or a region inside a file. In the former case, a lock attempt on a locked file will fail. In the latter case, an attempt to lock a region already locked will fail. In either case, the requesting process is not permitted to access the file since the lock has not been granted by the kernel.

Pipes and named pipes have a similar implementation, as their functionality is almost the same. The creation of process is different. However, in either case a file descriptor is returned which refers to pipe. Upon creation, one page of memory is associated with opened pipe. This memory is treated like circular buffer to which write operations are done atomically. When the buffer is full, the writing processes block. If a read request is made for more data than available, the reading processes block. Each pipe has a wait queue associated with it. Processes are added and removed from the queue during the read and writes.

Semaphores are implemented with wait queues and follow classical semaphore model. Each semaphore has an associated value. Two operations, up() and down() are implemented on the semaphore. When the value of the semaphore is zero, the process performing the decrement on the semaphore is blocked on the wait queue. Semaphore arrays are simply a contiguous set of semaphores. Each process also maintains a list of semaphore operations it has performed, so that if the process exits prematurely, these operations can be undone.

The message queue is a linear linked-list, to which processes read or write a sequence of bytes. Messages are received in the same order that they are written. Two wait queues are associated with the message queues, one for processes that are writing to a full message queue, and another for serializing the message writes. The actual size of the message is set when the message queue is created.

Shared memory is the fastest form of IPC. This mechanism allows processes to share a region of their memory. Creation of shared memory areas is handled by the memory management system. Shared pages are attached to the user processes virtual memory space by the system call sys_shmat(). A shared page can be removed from the user segment of a process by calling the sys_shmdt() call.

The Unix domain sockets are implemented in a similar fashion to pipes, in the sense that both are based on a circular buffer based on a page of memory. However, sockets provide a separate buffer for each communication direction.

12.10.4 Data Structures

In this section, the important data structures needed to implement the above IPC mechanisms are described.

Signals are implemented through the signal field in the task_struct structure. Each signal is represented by a bit in this field. Thus, the number of signals a version of Linux can support is limited to the number of bits in a word. The field blocked holds the signals that are being blocked by a process.

There is only one data structure associated with wait queues, the wait_queue structure. These structures contain a pointer to the associated task_struct, and are linked into a list.

File locks have an associated file_lock structure. This structure contains a pointer to a task_struct for the owning process, the file descriptor of the locked file, a wait queue for processes which are waiting for the cancellation of the file lock, and which region of the file is locked. The file_lock structures are linked into a list for each open file.

Pipes, both nameless and named, are represented by a file system inode. This inode stores extra pipe-specific information in the pipe_inode_info structure. This structure contains a wait queue for processes which are blocking on a read or write, a pointer to the page of memory used as the circular buffer for the pipe, the amount of data in the pipe, and the number of processes which are currently reading and writing from/to the pipe.

All system V IPC objects are created in the kernel, and each have associated access permissions. These access permissions are held in the ipc_perm structure. Semaphores are represented with the sem structure, which holds the value of the semaphore and the pid of the process that performed the last operation on the semaphore. Semaphore arrays are represented by the semid_ds structure, which holds the access permissions, the time of the last semaphore operation, a pointer to the first semaphore in the array, and queues on which processes block when performing semaphore operations. The structure sem_undo is used to create a list of semaphore operations performed by a process, so that they can all be undone when the process is killed.

Message queues are based on the msquid_ds structure, which holds management and control information. This structure stores the following fields:

- Access permissions
- Link fields to implement the message queue (i.e. pointers to msquid_ds)
- Times for the last send, receipt and change
- Queues on which processes block, as described in the previous section
- The current number of bytes in the queue
- The number of messages
- The size of the queue (in bytes)
- The process number of the last sender
- The process number of the last receiver.

A message itself is stored in the kernel with a msg structure. This structure holds a link field, to implement a link list of messages, the type of message, the address of the message data, and the length of the message.

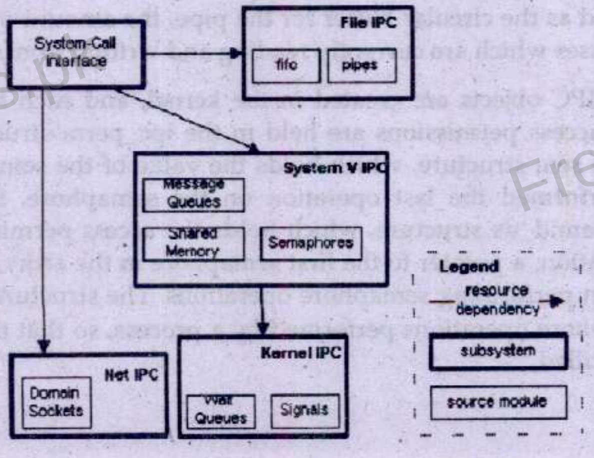
The shared memory implementation is based on the shmid_ds structure, which, like the msquid_ds structure, holds management and control information. The structure contains access control permissions, last attach, detach and change times, pids of the creator and last process to call an operation for the shared segment, number of processes to which the shared memory region is attached to, the number of pages which make up the shared memory region, and a field for page table entries.

The Unix domain sockets are based on the socket data structure, described in the Network Interface section.

12.10.5 Subsystem Structure

The following figure shows the IPC subsystem resource dependencies. Control flows from the system call layer down into each module. The System V IPC facilities are implemented in the ipc directory of the kernel source. The kernel IPC module refers to IPC facilities implemented within the kernel directory. Similar conventions hold for the File and Net IPC facilities.

The System V IPC module is dependant on the Kernel IPC mechanism. In particular, semaphores are implemented with wait queues. All other IPC facilities are implemented independently of each other.



12.10.6 Subsystem Dependencies

The figure shows the resource dependencies between the IPC subsystem and other kernel subsystems.

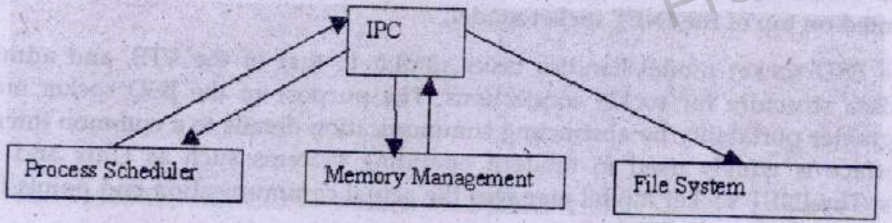


Figure: IPC Subsystem Dependencies

The IPC subsystem depends on the file system for sockets. Sockets use file descriptors, and once they are opened, they are assigned to an inode. Memory management depends on IPC as the page swapping routine calls the IPC subsystem to perform swapping of shared memory. IPC depends on memory management primarily for the allocation of buffers and the implementation of shared memory.

Some IPC mechanisms use timers, which are implemented in the process scheduler subsystem. Process scheduling relies on signals. For these two reasons, the IPC and Process Scheduler modules depend on each other.

12.11 Network Interface

12.11.1 Goals

The Linux network system provides network connectivity between machines, and a socket communication model. Two types of socket implementations are provided: BSD sockets and INET sockets. BSD sockets are implemented using INET sockets.

The Linux network system provides two transport protocols with differing communication models and quality of service. These are the unreliable, message-based UDP protocol and the reliable, streamed TCP protocol. These are implemented on top of the IP networking protocol. The INET sockets are implemented on top of both transport protocols and the IP protocol.

Finally, the IP protocol sits on top of the underlying device drivers. Device drivers are provided for three different types of connections: serial line connections (SLIP), parallel line connections (PLIP), and ethernet connections. An address resolution protocol mediates between the IP and ethernet drivers. The address resolver's role is to translate between the logical IP addresses and the physical ethernet addresses.

12.11.2 External Interface

The network services are used by other subsystems and the user through the socket interface. Sockets are created and manipulated through the socketcall() system call. Data is sent and received using read() and write() calls on the socket file descriptor.

No other network mechanism/functionality is exported from the network sub-system.

12.11.3 Subsystem Description

The BSD socket model is presented to the user processes. The model is that of a connection-oriented, streamed and buffered communication service. The BSD socket is implemented on top of the INET socket model.

The BSD socket model handles tasks similar to that of the VFS, and administers a general data structure for socket connections. The purpose of the BSD socket model is to provide greater portability by abstracting communication details to a common interface. The BSD interface is widely used in modern operating systems such as Unix and Microsoft Windows. The INET socket model manages the actual communication end points for the IP-based protocols TCP and UDP.

Network I/O begins with a read or write to a socket. This invokes a read/write system call, which is handled by a component of the virtual file system, (the chain of read/write calls down the network subsystem layers are symmetric, thus from this point forward, only writes are considered). From there, it is determined that the BSD socket sock_write() is what implements the actual file system write call; thus, it is called. This routine handles administrative details, and control is then passed to inet_write() function. This in-turn calls a transport layer write call (such as tcp_write()).

The transport layer write routines are responsible for splitting the incoming data into transport packets. These routines pass control to the ip_build_header() routine, which builds an ip protocol header to be inserted into the packet to be sent, and then tcp_build_header() is called to build a tcp protocol header. Once this is done, the underlying device drivers are used to actually send the data.

The network system provides two different transport services, each with a different communication model and quality of service. UDP provides a connectionless, unreliable data transmission service. It is responsible for receiving packets from the IP layer, and finding the destination socket to which the packet data should be sent. If the destination socket is not present, an error is reported. Otherwise, if there is sufficient buffer memory, the packet data is entered into a list of packets received for a socket. Any sockets sleeping on a read operation are notified, and awoken.

The TCP transport protocol offers a much more complicated scheme. In addition to handling data transfer between sending and receiving processes, the TCP protocol also performs complicated connection management. TCP sends data up to the socket layer as a stream, rather than as a sequence of packets, and guarantees a reliable transport service.

The IP protocol provides a packet transfer service. Given a packet, and a destination of the packet, the IP communication layer is responsible for the routing of the packet to the correct host. For an outgoing data stream, the IP is responsible for the following:

- Partitioning the stream into IP packets
- Routing the IP packets to the destination address
- Generating a header to be used by the hardware device drivers
- Selecting the appropriate network device to send out on

For an incoming packet stream, the IP must do the following:

- Check the header for validity
- Compare the destination address with the local address and forwarding it along if the packet is not at it's correct destination

- Defragment the IP packet
- Send the packets up to the TCP or UDP layer to be further processed.

The ARP (address resolution protocol) is responsible for converting between the IP and the real hardware address. The ARP supports a variety of hardware devices such as Ethernet, FDDI, etc. This function is necessary as sockets deal with IP addresses, which cannot be used directly by the underlying hardware devices. Since a neutral addressing scheme is used, the same communication protocols can be implemented across a variety of hardware devices.

The network subsystem provides its own device drivers for serial connections, parallel connections, and ethernet connections. An abstract interface to the various hardware devices is provided to hide the differences between communication mediums from the upper layers of the network subsystem.

12.11.4 Data Structures

The BSD socket implementation is represented by the socket structure. It contains a field, which identifies the type of socket (streamed or datagram), and the state of the socket (connected or unconnected). A field that holds flags, which modify the operation of the socket, and a pointer to a structure that describes the operations that can be performed on the socket, are also provided. A pointer to the associated INET socket implementation is provided, as well as a reference to an inode. Each BSD socket is associated with an inode.

The structure sk_buff is used to manage individual communication packets. The buffer points to the socket to which it belongs, contains the time it was last transferred, and link fields so that all packets associated with a given socket can be linked together (in a doubly linked list). The source and destination addresses, header information, and packet data are also contained within the buffer. This structure encapsulates all packets used by the networking system (i.e. tcp packet, udp packets, ip packets, etc.).

The sock structure refers to the INET socket-specific information. The members of this structure include counts of the read and write memory requested by the socket, sequence numbers required by the TCP protocol, flags which can be set to alter the behavior of the socket, buffer management fields, (for example, to maintain a list of all packets received for the given socket), and a wait queue for blocking reads and writes. A pointer to a structure that maintains a list of function pointers that handle protocol-specific routines, the proto structure, is also provided. The proto structure is rather large and complex, but essentially, it provides an abstract interface to the TCP and UDP protocols. The source and destination addresses, and more TCP-specific data fields are provided. TCP uses timers extensively to handle time-outs, etc. thus the sock structure contains data fields pertaining to timer operations, as well as function pointers, which are used as callbacks for timer alarms.

Finally, the device structure is used to define a network device driver. This is the same structure used to represent file system device drivers.

12.11.5 Subsystem Structure

The General Network contains those modules that provide the highest-level interface to user processes. This is essentially the BSD Socket interface, as well as a definition of the protocols supported by the network layer. Included here are the MAC protocols of 802.x, ip, ipx, and AppleTalk.

The Core services correspond to high-level implementation facilities, such as INET sockets, support for firewalls, a common network device driver interface, and facilities for datagram and TCP transport services.

The system call interface interacts with the BSD socket interface. The BSD socket layer provides a general abstraction for socket communication, and this abstraction is implemented using the INET sockets. This is the reason for the dependency between the General Network module and the Core services module.

The Protocol modules contain the code that takes user data and formats them as required by a particular protocol. The protocol layer finally sends the data to the proper hardware device, hence the dependency between the Network Protocols and Network Devices modules.

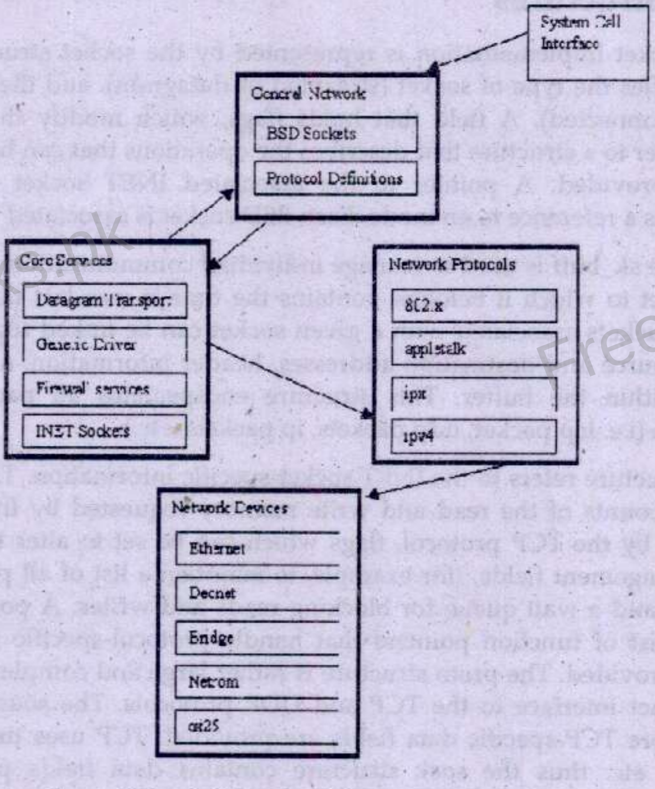

System Call Interface
General Network
BSD Sockets
Protocol Definitions
Core Services
Datagram Transport
Generic Driver
Firewall services
INET Sockets
Network Protocols
802.x
appletalk
ipx
ipv4
Network Devices
Ethernet
Decnet
Bridge
Netrom
ax25

Figure: Network Subsystem Structure

The Network devices module contains high-level device-type specific routines. The actual device drivers reside with the regular device drivers under the directory drivers/net.

12.11.6 Subsystem Dependencies

The network subsystem depends on the Memory Management system for buffers. The File System is used to provide an inode interface to sockets. The file system uses the network system to implement NFS. The network subsystem uses the kerneld daemon, and thus it depends on IPC. The network subsystem uses the Process Scheduler for timer functions and to suspend processes while network transmissions are executed.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

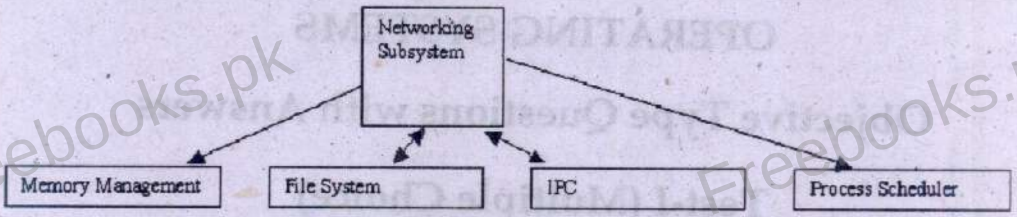


Figure: Dependencies between the networking subsystem and other subsystems.

Review Questions

1. What is the Free Software Foundation/GNU?

The Free Software Foundation is the principal organizational sponsor of the GNU Project. GNU developed many of the tools including C compiler that are part of Linux operating system.

2. What is Linux?

Linux is the name of an operating system kernel developed by Linus Torvalds. It has been expanded and improved by thousands of people on the Internet.

3. Why is Linux popular?

Linux is portable and is based on standards. It is written in C and can support many users. It can run multiple tasks simultaneously.

4. In which language is Linux written?

Most of Linux is written in C programming language. The portable style C programs can be moved from one platform to another by simply recompiling the code.

5. Why is Linux file system referred to as hierarchical?

The Linux filesystem is a hierarchy in which the root directory appears at the top of the system. The branches come from the root with each branch supporting one or more directory files.

OPERATING SYSTEMS

Objective Type Questions with Answers

Test-I (Multiple Choice)

1. **What is the name given to the organized collection of software that controls the overall operation of a computer?**
   a. Working system b. Peripheral system
   C. Operating system d. Controlling system
   Answer: c
2. **A computer cannot boot if it does not have the:**
   a. Compiler b. Loader c. Operating system d. Assembler
   Answer: c
3. **What is the name of the technique in which the operating system of a computer executes several programs concurrently by switching back and forth between them?**
   a. Partitioning b. Multitasking c. Windowing d. Paging
   Answer: b
4. **Operating system:**
   a. links a program with the subroutines it references
   b. provides a user-friendly interface
   c. enables the programmer to draw a flowchart
   d. all of the above
   Answer: b
5. **In what way is an operating system like a government?**
   a. It does not often function correctly
   b. It creates an environment within which other programs can do useful work.
   c. It performs most useful functions by itself
   d. It is always concerned primarily with the individual's needs.
   Answer: b
6. **The ability of operating system that allows effective development, testing, and introduction of new system function without affecting with other services is called:**
   a. Convenience b. Efficiency
   c. Ability to function independently d. Ability to evolve
   Answer: d
7. **Which of the following are services is provided by the operating systems?**
   a. Program creation and execution b. Internal and external error detection
   c. Providing access to hardware and resources d. All
   Answer: d
8. **Which of the following is considered a resource that may be allocated by operating system?**
   a. CPU b. Bus c. File system d. Instruction register
   Answer: a
9. **Embedded computers typically run on a ____ operating system:**
   a. Real-time b. Windows XP c. Network d. Clustered
   Answer: a
10. **Which of the following is NOT a resource that may be allocated by operating system?**
    a. CPU b. Storage device c. File system d. Memory
    Answer: c

11. **Which of the following is NOT an operating system?**

a. Linux b. Solaris c. Windows XP d. Oracle

Answer: d

12. **Which of the following is an example of a systems program?**

a. Command Interpreter b. Web Browser

c. Text Formatter d. Database system

Answer: a

13. **A system that supports multiple processes per user is called a:**

a. Multi-user system b. Multi-programming system

c. Multi-tasking system d. None of the above

Answer: c

14. **In which type of multiprocessing, a master processor schedules and allocates work to slave processors?**

a. Distributed multiprocessor systems b. Asymmetric multiprocessing

c. Symmetric multiprocessing d. Client-server system

Answer: b

15. **Which part of the operating system is responsible for CPU scheduling?**

a. Main memory manager b. I/O system manager

c. System call d. None of the above

Answer: d

16. **Which of the following is false about disk when compared to main memory?**

a. non-volatile b. longer storage capacity

c. lower price per bit d. faster

Answer: d

17. **The two modes of operation of an operating system are called:**

a. Process and Kernel b. Ready and running

c. Interrupt and system d. Kernel and user

Answer: d

18. **For a single-processor system:**

a. Processes spend long times waiting to execute

b. There will never be more than one running process

c. Input-output always causes CPU slowdown

d. Process scheduling is always optimal

Answer: b

19. **Process is:**

a. program in High level language kept on disk b. contents of main memory

c. a program in execution d. a job in secondary memory

Answer: c

20. **The dispatcher:**

a. Actually schedules the tasks into the processor b. Puts tasks in I/O wait

c. Is always small and simple d. Never changes task priorities

Answer: a

21. **System programs such as compilers are designed so that they are:**

a. Re-enterable b. Non reusable c. Serially usable d. Recursive

Answer: a

22. **A process is put into wait state when:**

a. It requests for a service that OS is not ready to perform

b. It requests for a resources that is busy

c. Waiting for a process to provide some input

d. All

Answer: d

23. **A scheduler that selects a swapped-out process and puts it into ready queue is known as:**

a. Short level scheduler b. Long level scheduler

c. Medium level scheduler d. None

Answer: c

24. When does a context switch occur?
a. When a process makes a system call
b. When an error occurs in a running program
c. When an external interrupt occurs
d. All
Answer: d

25. A process that has many short CPU bursts is called a:
a. CPU bound process
b. I/O bound process
c. I/O and CPU burst cycle
d. None of the above
Answer: b

26. An operating system module that performs context switching is called:
a. CPU scheduler
b. Dispatcher
c. Context switcher
d. None
Answer: b

27. A program is:
a. Another name for process
b. An active entity
c. Passive entity
d. All
Answer: b

28. A process control block should contain:
a. Process ID
b. Location to store register value
c. A list of all open files
d. All
Answer: d

29. The ____________ of a process contains temporary data such as function parameters, return addresses, and local variables.
a. Text Section
b. Data Section
c. Program Counter
d. Stack
Answer: d

30. The following are valid process states..
a. Next, Running, Halting
b. Terminating, Waiting, Threshing
c. Running, Blocked, Waiting
d. None
Answer: c

31. The list of processes waiting for a particular I/O device is called a(n) ____.
a. Standby queue
b. Device queue
c. Ready queue
d. Interrupt queue
Answer: b

32. Long-term scheduling is performed ________
a. Typically on submitted jobs
b. When processes must be moved from waiting to ready state
c. On processes in the ready queue
d. All
Answer: a

33. Medium-term scheduling is performed:
a. Typically on submitted jobs
b. When processes must be moved from waiting to ready state
c. On processes in the ready queue
d. All
Answer: b

34. ________ scheduling is approximated by predicting the next CPU burst with an exponential average of the measured lengths of previous CPU bursts.
a. Multilevel queue
b. Round Robin
c. FCFS
d. SJF
Answer: d

35. Assume there are 4 processes in the system running or ready to run. If round robin scheduling algorithm with a time quantum of 15 ms is used, then a waiting process will not wait more than __________:
a. 34 ms
b. 45 ms
c. 15 ms
d. 60 ms
Answer: b

36. In which of the following scheduling algorithms, starvation is possible?
a. Round robin
b. First come first serve

c. Multi-level queue scheduling d. All
Answer: c

37. The SJF algorithm executes first the job
a. That last entered the queue b. That first entered the queue
c. That has been in the queue the longest d. With the least processor needs
Answer: d

38. The ____ scheduling algorithm is designed especially for time-sharing systems.
a. Multilevel queue b. Round Robin c. FCFS d. SJF
Answer: b

39. Which of the following scheduling algorithms must be non-preemptive?
a. Priority algorithm b. Round Robin c. FCFS d. SJF
Answer: c

40. _______ takes a particular predetermined workload and defines the performance of each method for that workload.
a. Deterministic modeling b. Real-time scheduling
c. Queueing model d. Aging
Answer: a

41. An optimal job scheduling scheme is:
a. FCFS b. RR c. SJF d. Priority
Answer: c

42. The FIFO algorithm:
a. Executes first the job that last entered the queue
b. Executes first the job that first entered the queue
c. Execute first the job that has been in the queue the longest
d. Executes first the job with the least processor needs
Answer: b

43. Among CPU scheduling policies, First Come First Serve (FCFS) is attractive because:
a. It is simple to implement
b. It is fair to all processes
c. It minimizes the total waiting time in the system
d. It minimizes the average waiting time in the system
Answer: a

44. __________ is the number of processes that are completed per time unit:
a. CPU utilization b. Response time
c. Turnaround time d. Throughput
Answer: d

45. A circular queue is the most appropriate data structure for ______ scheduling.
a. RR b. FCFS c. SJF d. Multilevel
Answer: a

46. Turnaround time refers to the amount of time___________.
a. That CPU utilization is minimized b. Needed to execute particular process
c. A process has been waiting in the ready queue d. None
Answer: b

47. ___________ is the amount of time it takes from when a request was submitted until the first response is produced
a. CPU utilization b. Response time
c. Turnaround time d. Throughput
Answer: b

48. The threading model supported by Linux operating system is:
a. Many-to-one b. One-to-one- c. One-to-many d. Many-to-many
Answer: b

49. Which of the following types of resources can create a deadlock?
a. Shared b. Consumable c. Reusable d. All
Answer: a

50. In a system, necessary conditions to arise a deadlock situation if:
a. Any one of mutual exclusion, progress and bounded waiting occurs
b. Resource-allocation graph has cycles
c. Hold simultaneously mutual exclusion, hold-and-wait, non-preemption and circular wait
d. None
Answer: c

51. One necessary condition for deadlock is ______ that states that at least one resource must be held in a non-sharable mode.
a. Hold and wait b. Mutual exclusion c. Circular wait d. No preemption
Answer: b

52. Which of the following is most often used by operating systems to handle deadlocks?
a. Use protocols to prevent or avoid deadlocks b. Detect and recover from deadlocks
c. Pretend that deadlocks never occur d. None
Answer: c

53. Which of the following statement is true?
a. A safe state may lead to a deadlocked state
b. An unsafe state may lead to a deadlocked state
c. A safe state is a deadlocked state
d. None
Answer: b

54. A deadlocked state occurs when two or more processes are waiting indefinitely for an event that can be:
a. Occurred only by an I/O event
b. Happened by one of the ready processes
c. Caused only by one of the waiting processes
d. Caused only by a circular wait
Answer: c

55. A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes is called:
a. Starvation b. Deadlock state c. Hold and wait d. Circular wait
Answer: c

56. Which of the following statement is true for Banker's algorithm?
a. Can allocates multiple instances of each resource type
b. A deadlock prevention scheme
c. More efficient than the resource-allocation-graph scheme
d. Both a and c
Answer: a

57. The Banker's Algorithm is an example of a technique for:
a. Deadlock prevention b. Deadlock avoidance
c. Deadlock detection d. Deadlock recovery
Answer: b

58. If preemption is required to deal with deadlocks, then which statement must be addressed?
a. Preempt one process at a time until the deadlock cycle is eliminated
b. Preempt all deadlocked processes regardless of their cost and benefits
c. Select a victim process to be preempted with minimum cost; must roll back the process to safe state and restart; ensure starvation will not occur
d. Both a and b
Answer: c

59. A cycle in resource allocation graph ________ leads to a deadlock:
a. Always b. May not c. Must d. None
Answer: b

60. Inter-process communication:
a. is required for all processes b. is usually done via disk drives
c. is never necessary d. allows processes to synchronize activity

Answer: d

61. **A shared memory model is:**
   a. context switching
   b. a mechanism of storing process state information in process control block
   c. a inter-process communication model
   d. None
   Answer: c

62. **A semaphore:**
   a. can be modified simultaneously by multiple threads
   b. is accessed through only one standard operation
   c. is essentially an integer variable
   d. cannot be used to control access to a thread's critical sections
   Answer: c

63. **Which hardware device is used to map logical addresses into physical addresses?**
   a. Overlays
   b. Memory management unit
   c. Address translator
   d. Page table
   Answer: b

64. **The memory allocation scheme subject to external fragmentation is:**
   a. segmentation
   b. swapping
   c. pure demand paging
   d. multiple contiguous fixed partitions
   Answer: a

65. **Base register holds the:**
   a Smallest legal logical memory address
   b. Smallest legal physical memory address
   c. Lowest order logical memory address
   d. None
   Answer: b

66. **Paging is a memory management scheme that:**
   a Permits the logical address space of a process to be non-contiguous
   b. Avoids external fragmentation and the need for compaction
   c. Solves the considerable problem of fitting memory chunks of fixed sizes onto the back store.
   d. All
   Answer: b

67. **An address generated by a CPU is referred to as a:**
   a. Physical address
   b. Logical address
   c. Post relocation register address
   d. Memory-Management Unit (MMU) generated address
   Answer: b

68. **Which of the following dynamic storage-allocation algorithms results in the largest leftover hole in memory?**
   a. First fit
   b. Best fit
   c. Worst fit
   d. None
   Answer: c

69. **Which of the following is true of compaction?**
   a. It can be done at assembly, load, or execution time.
   b. It is used to solve the problem of internal fragmentation.
   d. It cannot shuffle memory contents.
   d. It is possible only if relocation is dynamic and done at execution time.
   Answer: d

70. **The purpose of a Memory Management Unit is to __________.**
   a. perform run-time mapping from virtual to physical addresses
   b. ensure protection of the memory space allocated to every process
   c. Both a and b are correct responses.
   d. None of these responses is correct
   Answer: a

71. **Which of the following statements is false?**
   a. A small page size causes large page tables
   b. Internal fragmentation is increased with small pages

c. A large page size causes instructions and data that will not be referenced brought into primary storage

d. I/O transfers are more efficient with large pages

Answer: b

72. A page fault:

a. is an error is a specific page

b. occurs when a program accesses a page of memory

c. is an access to a page not currently in memory

d. is a reference to a page belonging to another program

e. None of the above

Answer: c

73. The second-chance (clock) algorithm is an efficient approximation technique for:

a. LRU page replacement b. LFU page replacement

c. benchmarking file system performance d. benchmarking raw disk I/O

Answer: a

74. Implementing LRU precisely in an OS is expensive, so practical implementations often use an approximation called:

a. MRU b. MFU c. LFU d. None

Answer: d

75. The LRU algorithm:

a. Pages out pages that have been used recently

b. Pages out pages that have not been used recently

c. Pages out pages that have been least used recently

d. Pages out the first page in a given area

Answer: c

Test-II (Multiple Choice)

1. Which OS strategy popularized the idea personal productivity tools, bitmap graphics, and virtual terminals?

a. Batch processing system b. Timesharing system

c. Personal computers & workstations d. Real-timer systems

e. Network computer systems f. None of the above

g. All of the above

Answer: c

2. Which OS strategy incorporates a medium term scheduler?

a. Batch processing system b. Timesharing system

c. Personal computers and workstations systems d. Real-timer systems

e. Network computer systems f. None of the above

g. All of the above

Answer: a

3. Which OS strategy led to wide spread use of distributed computing approaches, such as the client-server protocol?

a. Batch processing system b. Timesharing system

c. Personal computers and workstations systems d. Real-timer systems

e. Network computer systems f. None of the above

g. All of the above

Answer: e

4. Which OS strategy provides an interactive virtual machine to each of its users?

a. Batch processing system b. Timesharing system

c. Personal computers and workstations systems d. Real-timer systems

e. Network computer systems f. None of the above

g. All of the above

Answer: b

5. **Which OS strategy assures its users that it will deliver service within a previously agreed upon time?**

a. Batch processing system
b. Timesharing system
c. Personal computers and workstations systems
d. Real-timer systems
e. Network computer systems

Answer: d

6. **Which of the following statements are true about multiprogramming?**

a. It is not used on contemporary personal computers.
b. It encourages the abstraction of process so that the OS can distinguish among different running programs.
c. It forces the OS to incorporate processor scheduling.
d. The use of threads makes multiprogramming obsolete.
e. It was very popular in the 1970s, but is rarely used in contemporary operating systems.

Answer: b, c

7. **Which OS strategies can use multiprogramming?**

a. Batch processing system
b. Timesharing system
c. Personal computers and workstations systems
d. Real-time systems
e. Network computer systems
f. All of the above

Answer: f

8. **The *turnaround time* for computer work is intuitively defined as the time from when the work is given to the computer until the time the request has been completed. In a batch system, the turnaround time is:**

a. The time from when the job is submitted, until the time the first command in the job specification is completed.
b. The time from when the medium term scheduler allocates memory until the time the job vacates memory.
c. The time from when the job is submitted, until the time the results of the job are returned to the user.
d. The time from when the job is submitted, until the time the job first is allocated the processor.

Answer: c

9. **How is a resource characterized in an operating system?**

a. The totality of allocatable hardware units.
b. Anything that a process/thread can request, and the process may be suspended if it is not available.
c. Memory, normally held by a parent process/thread.
d. The remaining CPU time allocation of a process/thread.
e. CPU time and primary memory.

Answer: b

10. **How is a file characterized in an operating system?**

a. A tool used to shape metal and wood.
b. The underlying storage mechanism for a database management system.
c. A DOS mechanism for referencing BIOS information.
d. A Unix mechanism that allows users to protect their persistent data.
e. A named, linear sequence of bytes.

Answer: e

11. **A file descriptor is:**

a. A resource abstraction used to implement icons on a desktop.
b. The user's mechanism for specifying actions to be performed on a file.
c. An OS data structure used to keep the state of a file.
d. An abstract data type for secondary storage devices.
e. A POSIX-specific data structure used by the kernel.

Answer: c

12. Which of the following are considered to be resources?

a. Processor b. Memory c. Keyboard d. User e. All of the above.

Answer: a, b, c

13. A process is:

a. The same as a program
b. Independent of the program concept
c. A program in execution.
d. Usually built into the computer hardware.

Answer: c

14. Which of the following are parts of a process implementation?

a. Abstract machine environment b. List of resources c. Code d. Data e. Status

Answer: all

15. A thread differs from a lightweight process in that:

a. Threads have a parent process.
b. Lightweight processes execute their own programs.
c. Threads share a process's resources but lightweight processes do not.
d. Threads run in user space, but lightweight threads run in kernel space.
e. There is no difference.

Answer: e

16. Which of the following are factors that influence the design of an OS?

a. Performance b. Protection and security c. Correctness
d. Maintainability e. Commercial factors

Answer: all

17. Which of the following factors might prevent a unit of functionality from being implemented in an OS?

a. The size of the functionality
b. The time it would take to execute the functionality.
c. The difficulty in maintaining the functionality
d. The breadth of applications that would use the functionality.
e. The difficulty in convincing application programmers that this would be the best thing for them.

Answer: b, d

18. Which of the following statements are true of a protection mechanism:

a. It is a tool that the OS provides to implement security policies.
b. All operating systems incorporate them.
c. They can be implemented in the OS.
d. They can be implemented in middleware.
e. They can be implemented in conventional user code.

Answer: a, c

19. Which of the following are basic functions in an OS?

a. Process and resource management b. Device management
c. Memory management d. File management e. Windows management

Answer: a, b, c, d

20. Which physical hardware is controlled by the device management part of an OS?

a. Processor b. Memory c. Network port d. Hard disk e. None

Answer: c, d

21. Which physical hardware is controlled by the file management part of an OS?

a. Processor b. Memory c. Network port d. Hard disk

Answer: b, d

22. Which functional parts of the OS are used to support segmented virtual memory?

a. Process and resource management
b. Device management
c. Memory management
d. File management
e. Database management

Answer: Primarily b, c, d

23. **A user mode program can invoke OS services by:**
a. System call b. Message passing c. Executing a trap instruction
d. Having system administrator permission e. All of the above.
Answer: a, b, c

24. **A memory word is defined by**
a. The "width" of the memory. b. Size of data path between CPU and memory.
c. Size of the target datum in the machine. d. Size of the general registers in the machine.
e. A computer grammarian.
Answer: b

25. **Which of the following is NOT a component on the device controller interface?**
a. Command register b. Status register c. Data register or buffer
d. Frames register e. All are components
Answer: d

26. **Which of the following statements are true of device drivers?**
a. Device drivers execute privileged instructions.
b. Device drivers can use polling or interrupts for I/O.
c. Device drivers encapsulate the details of the device controller interface.
d. A device driver exports an abstraction of the device controller interface.
e. All of the above.
Answer: e

27. **An interrupt is the same as a trap in that they both:**
a. Block the process currently using the CPU.
b. Cause the processor to switch to supervisor mode and branch to a handler.
c. Indicate the completion of an I/O operation.
d. Indicate a user space request for OS service.
e. None of the above
Answer: b

28. **An interrupt is raised when:**
a. An error occurs in the device. b. A user process requires OS service.
c. A device sets its busy flag and resets its done flag.
d. A device resets its busy flag and sets its done flag.
e. None of the above.
Answer: d

29. **An interrupt handler is OS software.**
a. True b. False c. Sometimes it is software, other times it is hardware.
Answer: a

30. **A context switch is:**
a. The procedure of moving the computer from one room to another.
b. Abstract machine environment operation that determines relative position of stack frames.
c. Procedure of saving one process's hardware state & restoring another process's hardware state on CPU.
d. The process' address space.
e. None of the above.
Answer: c

31. **A computer with interrupts usually includes instructions to enable/disable interrupts:**
a. To prevent overhead due to slow devices.
b. To enable the OS to ignore rogue devices.
c. To prevent race conditions in the kernel.
d. To synchronize the CPU operation with the power cycle.
e. None of the above.
Answer: c

32. **The device controller interface is manipulated by which of the following components?**
a. Process manager b. Memory manager c. Device driver
d. File manager e. Application programs

Answer: c

33. **The device driver's primary clients are:**

a. Process manager b. Memory manager c. Device driver
d. File manager e. Application programs

Answer: d, e

34. **Which of the following steps are parts of a write operation using polling?**

a. The application requests a write operation.
b. The device driver queries the status register to determine if the device is idle. If the device is busy, the driver waits for it to become idle.
c. The driver copies data from user space memory to the controller's data register(s).
d. Driver stores an output command into the command register, thereby starting the device.
e. Driver repeatedly reads status register while waiting for device to complete its operation.

Answer: All

35. **What is the purpose of the Device Status Table for interrupt-driven I/O?**

a. To mark the status of each device at boot time.
b. To save the status of a pending I/O operation.
c. To make the current status of each device available to the entire kernel.
d. To save manufacturer's serial and model number for each device.
e. None of the above.

Answer: b

36. **Direct memory access is**

a. Used to increase a systems performance.
b. More secure than indirect I/O.
c. Used to allow device controller to read and write primary memory without using the CPU.
d. Used only in devices that perform direct I/O.
e. None of the above

Answer: a, c

37. **Computers with direct memory access (DMA) normally have higher performance than those without it because of all of the following reasons:**

a. They do not use polling.
b. Interrupts are almost always vectored on these machines.
c. They avoid excessive data bus conflict.
d. The software has better control of the detailed device activity.
e. None of the above.

Answer: e

38. **Buffering is usually necessary for there to be CPU-device operation overlap.**

a. True b. False c. It depends on the nature of the device.

Answer: a

39. **Buffering is usually sufficient for there to be CPU-device operation overlap.**

a. True b. False c. It depends on the nature of the device.

Answer: b

40. **In conventional hard disks, seek time dominates latency time.**

a. True b. False c. It depends on the nature of the device.

Answer: a

41. **Which of the following tasks are performed by the process manager?**

a. Resource management b. Synchronization management
c. Deadlock management d. Processor scheduling e. All of the above

Answer: e

42. **The "initial process" is the first program executed after a computer is powered up.**

a. True b. False

Answer: a

43. **When a process is created, which of the following actions occur?**

a. The child process clones the parent process identification.
b. A process descriptor is allocated for the process.

c. The new process loads the program it intends to use.
d. The new process receives a resource allocation from the process that created it.
e. None of the above.

Answer: b

44. **After a newly created process has been made ready for execution, which of the following actions occur?**

a. It waits for user input.
b. It creates window for user interaction.
c. It competes with other processes for the CPU.
d. It voluntarily yields CPU to other processes.
e. None of the above.

Answer: c

45. **The process address space is:**

a. The set of all primary memory locations allocated to the process.
b. The set of all secondary memory locations allocated to the process.
c. The set of all locations used by a process to reference any resource.
d. The system memory that implements the heap.
e. None of the above.

Answer: c

46. **Which of the following items is NOT usually a part of a process descriptor?**

a. Internal process name
b. State
c. Protection domain
d. Grandparent process
e. List of file descriptors

Answer: d

47. **Which of the following items is NOT usually a part of a resource descriptor?**

a. External resource name
b. Total units
c. Available units
d. List of available units
e. List of blocked processes

Answer: a

48. **A resource is anything that can potentially block a process from executing.**

a. True
b. False

Answer: a

49. **Reusable resources differ from consumable resources in that:**

a. A consumable resource must be returned to the system after the process finishes using it.
b. A reusable resource must be returned to the system after the process finishes using it.
c. A consumable resource has far less intrinsic value than a reusable resource
d. There can be an unbounded number of reusable resources.
e. None of the above.

Answer: b

50. **Which of the following components is NOT used in implementing a scheduler?**

a. Ready list
b. Dispatcher
c. Interrupt handler
d. Context switcher

Answer: C

51. **What are possible causes for a process to relinquish the CPU?**

a. It has finished processing
b. It voluntarily relinquishes the CPU
c. It is forcefully removed from the CPU
d. It requests a resource
e. All of the above

Answer: e

52. **What problems can cause voluntary CPU sharing to fail?**

a. A process may fail to call yield ()
b. Interrupts may be disabled for inordinate amounts of time
c. The scheduler mechanism may deadlock
d. The scheduler policy may be ill-defined
e. None of the above.

Answer: a

53. **The interval timer device is:**

a. A virtual device
b. An unused option on most contemporary computers

c. A programmable device that produces periodic interrupts
d. A pseudonym for the hardware clock
e. None of the above

Answer: c

54. **Schedulers have been heavily studied over the last three decades for which of the following reasons?**

a. Designers of multiprogramming operating systems knew the scheduler's behavior was critical to the performance of the individual process's perspective.
b. Those designers believed the scheduler's behavior might be critical to the overall behavior of the system.
c. The methodologies used to study schedulers were well entrenched in operations research.
d. Schedulers were a nice theoretical computer science problem.
e. All of the above.

Answer: e

55. **The *wait time* for a process is**

a. The amount of time a process needs to be in the running state before it is completed.
b. The amount of time the process spends waiting in the ready state before its first transition to the running state.
c. The amount of time between the moments a process first enters the ready state and the moment the process exits the running state for the last time.
d. The time the process waits for all resources to be allocated to it.
e. None of the above.

Answer: b

56. **A scheduling policy may attempt to optimize on:**

a. CPU utilization b. Average wait time c. Average turnaround time
d. Average throughput rate e. All of the above

Answer: e

57. **Non-preemptive processor scheduling algorithms**

a. Rely on interrupts to multiplex the CPU
b. Rely on cooperative sharing mechanisms to multiplex the CPU
c. Are the preferred class of algorithms to support real-time computing
d. Are most widely used because of their simplicity
e. None of the above.

Answer: b

58. **Which of the following are true of FCFS non-preemptive scheduling?**

a. Minimizes wait time b. Maximizes throughput rate.
c. Maximizes equitability. d. None of the above e. All of the above

Answer: d

59. **Which of the following statements are true of (hard and soft) real-time scheduling?**

a. They minimize the average wait time.
b. They maximize CPU utilization.
c. They use multi level queues.
d. They are supported in Windows NT kernel.
e. None of the above.
f. All of the above.

Answer: e

60. **Using RR scheduling with a time quantum of 40 and no scheduling overhead, what is the average wait time for the example process set?**

a. 40 b. 74 c. 164 d. 188 e. None

Answer: b

61. **Using RR scheduling with a time quantum of 40 and no scheduling overhead, what is the average turnaround time for the example process set?**

a. 40 b. 74 c. 172 d. 258 e. None

Answer: e (it is 354)

62. **Using RR scheduling with a time quantum of 40 and 10 units of time for scheduling overhead (dispatching, context switching, etc.) what is the average wait time for the example process set?**

a. 40 b. 74 c. 86 d. 94 e. None

Answer: d

63. **Using RR scheduling with a time quantum of 40 and 10 units of time for scheduling overhead (dispatching, context switching, etc.) what is the average turnaround time for the example process set?**

a. 296 b. 354 c. 378 d. 442 e. None

Answer: d

64. **What is the effect of increasing the time quantum to an arbitrarily large number for Round Robin scheduling?**

a. The average wait time decreases b. The average turnaround time increases
c. The algorithm behaves exactly the same as nonpreemptive FCFS
d. All of the above e. None of the above

Answer: c

65. **What is a critical section?**

a. The heart of a decision support module.
b. A section of code executed by a process, which interacts with the execution of a critical section of code executed by a different process.
c. The part of the kernel that contains the essential element of software that makes the kernel be trusted software.
d. This is a decision point in any resource manager, where processes are selected (or not selected) for resource allocation.
e. None of the above.

Answer: b

66. **What is a race condition?**

a. A condition in which the order in which processes execute certain sections of code determines the outcome of a computation.
b. A condition in which a block of code must execute sufficiently fast to avoid system failure.
c. A computing environment for benchmarking system programs.
d. A condition in which hardware must surpass a threshold performance in order for the code to be correct.
e. None of the above.

Answer: a

67. **The critical section problem can be solved by disabling and enabling interrupts.**

a. True b. False c. Some instances of critical section problem can but not all.

Answer: a

68. **Deadlock can occur when**

a. A process intends to be in two or more critical sections at a time.
b. One process holds a critical section, while asking for another.
c. Both a and b

Answer: c

69. **An acceptable solution to the critical section problem is required to meet which of the following constraints?**

a. Only one process at a time should be allowed to be in its critical section.
b. If a critical section is free and a set of processes indicates a need to enter into the critical section, then only those processes competing of the critical section participate in the selection of the process to enter the critical section.
c. Once a process attempts to enter its critical section, it cannot be postponed indefinitely even if no other process is in its critical section.
d. After a process requests entry into its critical section, only a bounded number of other processes may be allowed to enter their related critical sections before the original process enters its critical section.

e. All of the above.
f. None of the above.
Answer: e

70. A semaphore is:
a. A nonnegative integer variable
b. Accessed only by two access routines
c. Incremented by the V function
d. Decremented by the P function
e. All of the above
f. None of the above
Answer: e

71. The problem with disabling interrupts to implement semaphores is:
a. It is overkill
b. It interferes with other groups of processes that are synchronizing access to their own critical sections.
c. It may cause excessively long periods of times in which I/O completions are lost
d. All of the above
e. None of the above
Answer: d

72. An active implementation of semaphores differs from a passive implementation in that:
a. Active implementations call the scheduler on a P operation
b. Active implementations call the scheduler on a V operation
c. Active implementations are implemented with their own thread
d. Active implementations poll the semaphore
e. All of the above
Answer: b

73. Which of the following statements are true of monitors?
a. A monitor is an abstract data type
b. Only one process may be executing monitor functions at a time
c. Monitors cannot solve any synchronization problems that could not be solved using semaphores
d. All of the above
e. None of the above
Answer: d

74. Which of the following are benefits of using monitors?
a. There is an opportunity to control how a process or thread uses the data type, since has a specific interface.
b. Detailed manipulation of the data structure is centralized in one software module, thereby making it possible to prove properties about the behavior of the data structure.
c. Other software cannot take advantage of the implementation but instead must rely on the interface.
d. The implementation can evolve without the users of the data structure
e. All of the above
f. None of the above
Answer: e

75. Condition variables are required to augment monitors because:
a. The monitor language is not Turing complete unless it includes condition variables.
b. Having condition variables greatly simplifies the monitor function declarations
c. The monitor function may discover that it cannot proceed until some other process takes some particular action on the information protected by the monitor.
d. All of the above
e. None of the above
Answer: c

76. The solution to the Dining Philosophers problems demonstrates that monitors solve problems that cannot be solved with AND synchronization.
a. True
b. False
Answer: b

77. The disadvantage to implementing mailboxes in kernel space is:

a. Reuse b. Security c. More work/responsibility for the kernel
d. There are no disadvantages e. None of the above

Answer: c

78. Which of the following are approaches to handling deadlock?

a. Prevention b. Avoidance c. Detection and recovery
d. Manual deadlock management e. All of the above f. None of the above

Answer: e

79. Which of the following are necessary conditions for deadlock?

a. Mutual exclusion b. Hold and wait condition c. Circular wait condition
d. No preemption e. All of the above f. None of the above

Answer: e

80. In the deadlock prevention strategy,

a. The banker's algorithm is the best known implementation
b. Idea is to ensure that at least one of the necessary conditions for deadlock is false at all times
c. An ounce of this is worth a pound of recovery d. None of the above

Answer: b

81. In the deadlock avoidance strategy,

a. The banker's algorithm is the best known implementation
b. The resource manager avoids deadlock by staying in safe states
c. Each process makes maximum claim to memory manager when process begins to run
d. All of the above e. None of the above

Answer: d

82. In the detection and recovery strategy,

a. The banker's algorithm is the best known implementation
b. The detection phase can be executed periodically or sporadically
c. Recovery will necessarily terminate all active processes
d. All of the above e. None of the above

Answer: b

83. The system deadlock model

a. Has same number of states as total of the number of processes and the number of resources.
b. Is used to capture the collective process states
c. Is used to capture the collective resource states
d. All of the above e. None of the above

Answer: e

84. Process p_i is blocked in S_j if

a. pi can only release resources it holds b. pi cannot cause a transition out of Sj
c. There is another process pk that is deadlocked d. All of the above
e. None of the above

Answer: b

85. Process P_i is deadlocked in S_j if

a. Pi can only release resources it holds b. Pi cannot cause a transition out of Sj
c. Pi cannot cause a transition out of any Sk reachable from Sj
d. All of the above e. None of the above

Answer: c

86. In a prevention strategy, the hold and wait condition can be violated by

a. Requiring that a process request all resources it will use when it is created
b. Requiring a process to release all resources it currently holds before it request any new ones
c. it cannot be violated d. None of the above

Answer: a, b

87. In a prevention strategy, the circular wait condition can be violated by

a. Requiring that a process request all resources it will use when it is created
b. Requiring a process to release all resources it currently holds before it request any new ones

c. Establishing a total order on all resources, then forcing each process to acquire resources according to the total order

d. All of the above e. None of the above

Answer: d

88. In the context of deadlock avoidance, once a process enters an *unsafe* state, it will eventually deadlock

a. True b. False

Answer: b

89. In the avoidance strategy,

a. State transitions are permitted only when the system is certain that a deadlock cannot occur due to subsequent requests

b. The system is only allowed to transition to safe states

c. Each process must provide its maximum claim when it is created

d. All of the above e. None of the above

Answer: d

90. Which of the following are basic requirements that drive the OS memory manager designs?

a. The primary memory access time must be as small as possible.

b. The primary memory must be as large, logically, as possible.

c. The primary memory must be as small, physically, as possible

d. The primary memory must be cost-effective. e. All of the above. f. None

Answer: a, b, d

91. Which of the following are functions of the memory manager?

a. Define the process's address space. b. Allocate primary memory space to processes.

c. Map the process address space into the allocated portion of the primary memory.

d. Minimize access times using a cost-effective amount of primary memory.

e. All of the above. f. None of the above.

Answer: b, c, d

92. The result of compile time processing is:

a. Relocatable object modules b. Absolute object modules

c. Executable in-memory image d. All of the above e. None of the above

Answer: a

93. The result of link time processing is:

a. Relocatable object modules b. Absolute object modules

c. Executable in-memory image d. All of the above e. None of the above

Answer: b

94. The result of load time processing is:

a. Relocatable object modules b. Absolute object modules

c. Executable in-memory image d. All of the above e. None of the above

Answer: c

95. In a fixed-partition memory strategy, what causes internal fragmentation?

a. High stress computation

b. Mismatches between memory requests and the size of the memory partitions.

c. An inordinate number of thread stacks in one address space (region).

d. None of the above e. All of the above

Answer: b

96. Fixed-partition memory allocation strategies will call the loader far fewer times than will variable-partition memory allocation strategies.

a. True b. False

Answer: a

97. Fixed-partition memory allocation is best suited to which of the following types of operating systems?

a. Batch processing system b. Timesharing system

c. Personal computers and workstations systems d. Real-timer systems

d. Network computer systems e. All of the above

Answer: a

98. Fixed-partition memory allocation minimizes internal fragmentation.

a. True b. False

Answer: b

99. Variable-partition memory allocation minimizes internal fragmentation:

a. True b. False

Answer: a

100. Why is compaction so costly in a classic variable-partition memory allocation strategy?

a. Compaction is an O (n2) operation.

b. The system call overhead plus context switching sets the cost too high.

c. System must have the loader relocate each address every time an address space is moved.

d. All of the above.

e. None of the above.

Answer: c

101. With dynamic address relocation, the relocation register contains

a. The physical address of the first address in the address space

b. The virtual address issued by the CPU c. The linear offset into the address space

d. The size of the address space e. None of the above

Answer: a

102. With dynamic address relocation, the limit register contains

a. The physical address of the first address in the address space

b. The virtual address issued by the CPU c. The linear offset into the address space

d. The size of the address space e. None of the above

Answer: d

103. A virtual memory system, a process can use the CPU when only part of the address space is loaded into the primary memory.

a. True b. False

Answer: a

104. Which of the following are characteristics of a page table?

a. It varies with time b. It performs dynamic address translation

c. It maps the virtual address space into the physical address space

d. All of the above e. None of the above

Answer: d

105. Which of the following characteristics are differences between paging and segmentation?

a. They have different strategies for transfer block sizes

b. Segmentation is programmer visible, but paging is transparent to the programmer

c. Paging uses dynamic address translation, but segments are bound at link time.

d. All of the above e. None of the above

Answer: a, b

106. Paging transfers fixed-sized blocks between secondary and primary memory.

a. True b. False

Answer: a

107. Segmentation transfers fixed-sized blocks between secondary and primary memory.

a. True b. False

Answer: b

108. What is the purpose of the fetch policy?

a. It decides when a page should be loaded into primary memory

b. It determines the page to be removed from primary memory if all page frames are full

c. It determines where the fetched page should be loaded in primary memory

d. All of the above e. None of the above

Answer: a

109. What is the purpose of the replacement policy?

a. It decides when a page should be loaded into primary memory

b. It determines the page to be removed from primary memory if all page frames are full

c. It determines where the fetched page should be loaded in primary memory
d. All of the above e. None of the above
Answer: b

110. What is the purpose of the placement policy?
a. It decides when a page should be loaded into primary memory
b. It determines the page to be removed from primary memory if all page frames are full
c. It determines where the fetched page should be loaded in primary memory
d. All of the above e. None of the above
Answer: c

111. In an LRU implementation, there *reference bit* is
a. Used to indicate that a page has been referenced since the last page fault
b. Used to estimate the virtual time at which a page was last written
c. Cleared on each page fault by the process
d. All of the above e. None of the above
Answer: d

112. Practical measurements of paging system performance shows that for nearly all reference streams, thrashing will occur for memory allocations les than half the size of the virtual address space.
a. True b. False c. The results were inconclusive
Answer: a

113. A process's *working set*
a. Is the set of pages in the process's locality b. Contracts and expands according to locality
c. Is estimated using a window on the reference stream
d. All of the above e. None of the above
Answer: d

114. The working set is susceptible to thrashing if the window size is too small.
a. True b. False
Answer: a

115. A clock algorithm, used to implement the working set algorithm, is a global LRU algorithm.
a. True b. False
Answer: a

116. Which of the following are characteristics of all files?
a. It has a name b. It is a collection of bytes
c. It is stored on a disk d. All of the above e. None of the above
Answer: d

117. Which of the following are true for a file manager?
a. It maps a logical view onto blocks in a storage device
b. It allocates/deallocates blocks for a file c. It provides directories as collections of files
d. All of the above e. None of the above
Answer: d

118. Indexed sequential files are composed of an unordered set of sequential records.
a. True b. False
Answer: b

119. Which of these fields are typically in an external file descriptor?
a. External name b. Owner c. File pointer
d. Protection settings e. Storage device details
Answer: a, b, d, e

120. Which of these fields are typically in an internal file descriptor?
a. External name b. Owner c. File pointer
d. Protection settings e. Storage device details
Answer: all of above

121. A criticism of indexed allocation is that
a. Seek operations are slow b. A sequential read pass over the file is slow
c. It suffers from table fragmentation d. All of the above e. None of the above

Answer: c

122. Which of the following techniques are used for keeping track of the unallocated blocks on the storage device?

a. Heap b. Stack c. Linked list d. Bitmap e. All f. None

Answer: c, d

123. Which of the following operations are implemented by the directory management portion of the file manager?

a. Enumerate b. Copy c. Rename d. Delete e. Traverse

Answer: all of them

124. The ultimate range of abilities of a security policy is independent of the underlying protection mechanism.

a. True b. False

Answer: b

125. The way user authentication is implemented in an OS is determined by the

a. Protection mechanism b. Security policy
c. Cryptography engine d. Defibrillator e. None of the above

Answer: a

126. Whether or not a user has access to a particular file is determined by

a. Protection mechanism b. Security policy
c. Cryptography engine d. Flux capacitor e. None of the above

Answer: b

127. The parameter sharing security problem is characterized by

a. Critical section violations
b. Increased probability of non-determinate computation
c. A devious way for one process to write into another process's address space
d. All of the above e. None of the above

Answer: c

128. The confinement problem refers to security violations including

a. Failed authentication b. Information leaks out of a protected address space
c. Processes are unable to communicate with other processes
d. All of the above e. None of the above

Answer: b

129. The Trojan horse problem is characterized by

a. UCLA students stealing Tommy Trojan's horse
b. Packaging unauthorized software in a gift
c. A strategy for breaking security patterned after a chess move involving the knight
d. All of the above
e. None of the above

Answer: b

130. A protection state is characterized by

a. It represents the accessibility of resources by processes
b. It represents the accessibility of resources by threads
c. It represents the accessibility of objects by subjects
d. It represents the accessibility of subjects by objects
e. None of the above

Answer: c

Test-III (Multiple Choice)

1. An operating system is a

a. Set of user b. Form of Time-Sharing c. Set of programs d. Supervisor Program

Answer: c

2. In multiprogramming, two or more programs can be executed

a. By optimizing compilers b. With two computers c. Simultaneously d. Concurrently

Answer: d

3. Time-sharing of resources by users is

a. Based on time slices b. Based on input c. Event-driven d. operated by spooling

Answer: a

4. Management of an operating system is handled by

a. An interpreter b. Utility program c. The supervisor program d. CPU

Answer: c

5. The process of allocating main memory to programs and keeping the programs in memory separate from one another is called

a. Memory protection b. Virtual storage c. Memory management d. Real storage

Answer: c

6. The Windows version especially planned for strong stability and security

a. XL b. OL c. NT d. NS

Answer: C

7. The technique in shared systems that avoids interspersed printout from several programs is

a. Paging b. Slicing c. Queuing d. Spooling

Answer: d

8. The technique whereby part of the program is stored on disk and is brought into memory for execution as needed is called

a. Memory allocation b. Virtual storage c. Interrupts d. Prioritized memory

Answer: b

9. In multiprogramming, the process of confining each program to certain defined limits in memory is called

a. Spooling b. Program scheduling c. Time-sharing d. Memory protection

Answer: d

10. The corresponding memory spaces for pages are called

a. Page utilities b. Page blocks c. Page frames d. Page modules

Answer: c

11. List of programs waiting to be run are in

a. Page frames b. Shells c. The background d. Queues

Answer: D

12. Another name for virtual memory is

a. Virtual storage b. Background c. Foreground d. Utility

Answer: a

13. Which of the following types of binding take place during runtime?

a. Symbolic names to virtual addresses b. Virtual to physical addresses c. Both a and b

Answer: b

14. Which of the following does/do mapping of virtual to physical addresses?

a. Compiler b. linkage editor c. runtime VM sub-System

d. A and B e. B and C

Answer: c

15. In a system with paging only (every program is a collection of Pages, no segments), which of the following is PRESENT:

a. External fragmentation b. internal fragmentation

Answer: b

16. A system, which uses segmentation alone (NOT combined with paging), will suffer from

a. External fragmentation b. internal fragmentation

Answer: a

17. Reentrant code is

a. Code, which modifies itself b. code, which does not modify itself

c. Code with loops d. none of the above

Answer: b

18. What is binding?

a. A process for setting the actual addresses used by each un-relocatable address of a program

b. A process for loading segments A into the same area of memory as other routines B

c. c. A process for loading a routine only when it is called and not yet in memory.

Answer: a

19. What is a relocation register?

a. A base register used to give the limit value for a process.

b. A base register used to give the lowest physical address for a process.

c. A base register used to give the page offset of a page table for a process.

Answer: b

20. What is swapping?

a. Copying a process from one memory location to another to allow space for other processes

b. Copying process from disk to memory to provide page-in service

c. Copying process from memory to disk to allow space for other processes

Answer: c

21. Which memory segment allocation method is the poorest?

a. First-fit b. Best-fit c. Worst-fit

Answer: c

22. Why is there a reference bit?

a. To reference that a page was read b. To reference that a page was modified

c. To reference that a page was accessed

Answer: c

23. Ideally, what criteria should we use to replace pages?

a. Choose the victims to achieve the highest memory utilization.

b. Choose the victims to achieve the lowest page-fault rate.

c. Choose the victims to achieve the highest disk utilization.

Answer: b

24. What is trashing?

a. State where the system swaps out un-referenced pages.

b. State where the system decides that the degree of multiprogramming must be increased.

c. State where the system spends an excessive amount of time on paging, compared to the execution of processes.

Answer: c

25. Which page replacement algorithm suffers from Belady anomaly?

a. Least Recently Used b. FIFO c. Second Chance

Answer: b

26. Once the system detects trashing, what can the system do to eliminate this problem?

a. Increase the number of CPU-bound jobs b. Decrease the degree of multiprogramming

c. Install a faster CPU d. None of the above

Answer: b

27. Why is segmentation with paging considered?

a. To improve on internal fragmentation of segmentation

b. To improve on external fragmentation of segmentation

c. To avoid Belady's anomaly

Answer: b

28. Which of the following tells the system whether not a paper needs to write to the disk when it is replaced.

a. Base register b. Dirty bit c. Page table d. Limit register

Answer: b

29. Which of the following algorithms is the fastest?

a. First fit b. next fit c. Best-fit d. Worst fit

Answer: a

30. The idea of overlays is to keep an entire program in memory so that all data can be accessed quickly

a. True b. False

Answer: b

31. LRU replacement suffers from Belady's anomaly

a. True b. False

Answer: b

32. Compaction may be used to eliminate external fragmentation

a. True b. False

Answer: a

33. Which of the following is not a problem associated with contiguous allocation of disk space for a file?

a. External fragmentation of disk space b. frequent copying c. Random access

Answer: c

34. Which of the following methods would you choose if the file requires frequent direct access and also external fragmentation is to be avoided (to keep disk utilization high) ?

a) Linked allocation b) contiguous allocation c) indexed allocation

Answer: c

35. Symbolic Links can span across

a) One file system b) Multiple file systems

Answer: b

36. Difference between hard links and symbolic links is that the hard links allow for leaving these links dangling when the file entry is deleted, and symbolic links don't allow this.

a) True b) False

Answer: b

37. The file-reference count is used for

a) Counting number of bytes read from the file b) Counting number of open files

c) Counting number of links pointing to a file

Answer: c

38. Polling means that

a) The device controller reads a busy bit from the host

b) The host reads a status register from the device controller

c) The device controller waits for a signal from the device

Answer: b

39. The CPU detects an interrupt by

a) Using a busy bit b) Using an interrupt handler c) Using an interrupt request line

Answer: c

40. Cycle stealing happens

a) During the handshaking protocol between DMA controller and device controller;

b) During the handshaking protocol between the host and the device controller;

c) During the DMA data transfer between the main memory and the device controller

Answer: c

41. Character-stream devices allow for the following system calls:

a) Read, write, select b) get, put c) get, put, seek d) Read, write, seek

Answer: b

42. A successful blocking I/O system read call returns with

a) The full number of bytes requested b) Fewer number of bytes than requested c) No data

Answer: a

43. The difference between blocking and asynchronous read system call is that

a) Asynchronous call returns immediately and non-blocking does not return immediately

b) Asynchronous call returns immediately and gets complete results in the future, where non-blocking call returns immediately with whatever data available,

c) Asynchronous call returns immediately with whatever data available and the non-blocking call returns later with complete results

Answer: b

44. Spooling is required for devices such as

(a) Disks (b) network adapters (c) printers

Answer: c

45. Caching and buffering are always the same functions.

a) True b) false

Answer: b

46. The access method to a network device such as the network adapter is

a) Sequential b) random c) indexed

Answer: a

47. Which of the following is not a true about a log file system

a) Speeds up writes b) requires garbage collection

c) May slow down large read operations d) Saves disk space

Answer: d

48. What is the primary advantage of DMA?

a) It facilitates uniform naming b) it reduces the workload on the main CPU

c) It processes requests faster then the main CPU d) It can use the bus in burst mode

Answer: b

49. Which data structure is the best representation of a file system with shared files?

a) Tree b) directed acyclic graph c) directed graph

Answer: b

50. Which is not the necessary condition of a deadlock?

a. Mutual exclusion b. Hold and wait c. No preemption

d. Circular wait e. None of above

Answer: e

51. Which of the memory allocation schemes are subject to internal Fragmentation?

a. Multiple Contiguous Fixed Partitions b. Multiple Contiguous Variable Partitions

c. Segmentation d. None of above

Answer: a

52. With paging, the internal fragmentation is possible when:

a. Page does not quite fit the frame b. The last page of the job is less than maximum page size

c. The cache memory assigned to the page table entry is not the same as normal memory assigned to the page table entry

d. There is no such thing as internal fragmentation with paging

Answer: b

53. The scheduler that brings processes in memory and swaps them out on disk as needed is called:

a. Short-term scheduler b. Long-term scheduler c. Medium-term scheduler d. None

Answer: c

54. A data structure that contains information about processes currently blocked for I/O is called:

a. I/O Data Control Block b. File Control Block (FCB).

c. Process Control Block (PCB) d. Device Control Block (DCB).

Answer: a

55. For efficiency, the long-term scheduler selects:

a. A group of I/O bound jobs and then a group of CPU-bound jobs

b. A group of CPU-bound jobs and then an group of I/O-bound jobs

c. A random and even mix of CPU-bound and I/O-bound jobs

d. None of the above, since the long-term scheduler cannot

e. Influence efficiency of a multiprogrammed system

Answer: c

56. In an operating system a utility which lets the users issue and execute commands from the keyboard is called:

a. Terminal Handler b. Command Interpreter c. Kernel d. None

Answer: b

57. In MS-DOS systems, which of the following system programs interprets the user commands?

a. COMMAND.COM b. AUTOEXEC.BAT c. CONFIG.SYS d. MSDOS.EXE

Answer: a

58. CPU Scheduling algorithms are used for:

a. Choosing the next process to run from the PCBs in the ready list

b. Putting to sleep and waking up processes in an efficient manner

c. Allocating memory to the processes in a fair and efficient way

Answer: a

59. Which is NOT a primary reason for providing an environment that allows process cooperation:

a. Information sharing b. Computational speedup

c. Convenience d. Concurrency e. Modularity

Answer: d

60. Which scheduler is responsible for controlling the degree of Multiprogramming?

a. Long-term scheduler b. Medium-term scheduler

c. Short-term scheduler d. None of the above

Answer: a

61. The primary difference(s) between user-level threads and kernel threads is/are:

a. User level threads do not use OS services via system calls. kernel threads require system calls

b. User level threads are independent of each other, whereas Kernel threads can write into each other's memory space

c. User level threads require memory management where kernel threads do not

d. Both (a) and (b)

Answer: a

Test-IV (Multiple Choices)

1. The operating system does each of the following except

a. Allocates the computer's components to different programs

b. Synchronizes individual programs' activities

c. Ensures that programs terminate their execution

d. Provides the general mechanisms that are needed so that the programs execute in perfect harmony

Answer: a

2. Which is not an example of system software?

a. Command line interpreter b. Database management system

c. Window system d. Personal productivity package

Answer: d

3. An operating system is distinguished from other system software by each of the following except

a. It interacts directly with the hardware to provide an interface used by other system Software and application software

b. It allows different applications to share the hardware resources through its resource management policies

c. It can be used to support a broad range of application domains

d. The hardware resource abstractions it provides are convenient, but their use by applications is optional

Answer: d

4. Operating system functions are normally categorized into one of these categories except

a. Process, thread and resource management b. Memory management

c. Device management d. Window management

Answer: d

5. The operating system device manager handles all of the following according to policies

a. chosen by the designer or administrator except b. Specification of devices

c. Allocation of devices d. Sharing of devices

Answer: b

6. The process manager deals with the following except

a. Thread management b. Resource management

c. Process management d. Window management

Answer: d

7. Primary operating system design issues include the following except

a. Efficiency in the use of machine resources b. Compact memory representation

c. Resource isolation d. Maximizing availability of resource for use by applications

Answer: c

8. Basic operating system implementation mechanisms to address design issues do not include

a. Processor modes b. Kernels

c. Compilers d. Methods for invoking system services

Answer: c

9. Supervisor mode instructions are called the following, except

a. Privileged b. Protected c. Supervisor d. Super-user

Answer: d

10. Which of the following statements about kernels is not true?

a. kernel implements basic mechanism that assure secure operation of the entire operating system.

b. Kernel extensions must execute in supervisor mode.

c. The kernel executes as trusted software.

d. Kernel-implemented functions may be easy to implement.

Answer: b

11. The process manager commonly interacts with other components except

a. Device controller b. Device manager c. Memory manager d. File manager

Answer: a

12. Which of the following statements about multiprogramming is untrue?

a. Multiprogramming is essential in contemporary systems

b. I/O operations are ordinarily somewhat slower than CPU operations

c. Multiprogramming allows the system to support multiple window interfaces

d. Fundamental operation of modern systems depends on multiprogramming

Answer: b

13. Which is the least common reason that a running thread might cease using the CPU?

a. Thread completes execution b. Thread requests resource, and blocks

c. Thread voluntarily releases CPU d. Thread involuntarily releases CPU

Answer: a

14. Scheduling policy criteria often include the following except

a. Predictable performance b. Equitable sharing

c. Reducing context switching overhead d. Optimizing performance for a particular class of threads

Answer: c

15. Non-preemptive scheduling strategies that commonly rely on process/thread execution time include the following except

a. FCFS b. SJN c. Priority d. Deadline

Answer: a

16. The most commonly used preemptive scheduling algorithm is

a. Round robin b. SJN c. Priority d. Multi-level queueing

Answer: a

17. General approaches to handling deadlock do not include

a. Prevention b. Detection and recovery c. Deterrence d. Avoidance

Answer: c

18. Which of the following is not a standard condition on the way processes use resources?

a. Hold & wait b. Circular waiting c. No preemption d. Concurrent resource usage

Answer: d

19. Each of the following applies to deadlock avoidance except

a. Relies on ability to predict effect of satisfying allocation requests

b. Commonly used in modern operating systems

c. Tends to underutilize resources

d. Inherently conservative strategy

Answer: b

20. Each of the following applies to the detection and recovery strategy except

a. Difficult to determine when to execute the detection algorithm

b. The recovery phase preempts resources from processes

c. Manual determination of conditions to invoke detection

d. Processes cooperate with system in detection/recovery

Answer: d

21. A serially reusable resource graph can be reduced if the following conditions are met, except

a. The graph is acyclic b. Process p is not blocked

c. Process p has no request edges d. Assignment edges directly into p exist

Answer: a

22. Which statement is true?

a. Detection/recovery strategies are more conservative than avoidance strategies

b. Avoidance algorithms avoid only unrecoverable states

c. Detection/recovery ignores distinction between safe and unsafe states

d. Detection/recovery determine if any sequence of transitions would result in all processes becoming unblocked

Answer: b

23. Steps for statically binding addresses include the following except

a. Compile time translation b. Link time combining of relocatable object modules

c. Load time adjustment of the load module d. Run time rebinding

Answer: d

24. Fixed partition memory allocation strategies include the following except

a. Best fit b. Worst fit c. Any fit d. Best utilization

Answer: b

25. Variable partition memory allocation strategies include the following except

a. Fast fit b. Best fit c. Worst fit d. First fit

Answer: a

26. Which of the following statements about swapping is untrue?

a. Swapping is well suited to timesharing systems

b. Swapping is easy, with or without relocation hardware

c. A process should release memory if it's not going to execute soon

d. Swapping transfers the entire process image to secondary memory

Answer: b

27. Which is not a barrier to be overcome when implementing virtual memory?

a. Memory manager must be able to partition address space

b. System must be able to load a part of the address space anywhere in physical memory

c. System must be able to dynamically bind logical addresses to physical addresses

d. The system must limit the number of partitions for good performance

Answer: d

28. Which statement about segmentation is incorrect?

a. Variable-sized segments are defined by the programmer

b. Segments may be defined explicitly by language directives

c. Segments may be defined implicitly by program semantics

d. Segmentation systems are not subject to external fragmentation

Answer: d

29. Which statement about paging is untrue?

a. Paging uses single component addresses

b. The virtual address space is divided into variable-sized pages

c. Page boundaries are completely transparent to the programmer

d. Virtual memory manager is completely responsible for choosing the page to be moved back and forth between primary and secondary memory

Answer: b

30. Which statement about segmentation is false?

a. Segmentation provides explicit program control over the units of transfer in the memory system

b. Segments can be more efficient than paging

c. The virtual memory system has more difficulty placing segments, rather than pages, in memory

d. Segmentation (vs. paging) is rarely better suited to the behavior of processes

Answer: d

31. Which of the following is not a stack algorithm?

a. LRU b. LFU c. FIFO d. Random

Answer: d

32. The following statements about implementing working set algorithms are true except

a. The clock algorithm behaves like a global LRU algorithm

b. Clock algorithms have better fault rate performance than strict working set algorithms

c. Working set algorithm is harder to implement than LRU

d. Clock algorithms arrange all page frames in a circular list

Answer: b

33. Which statement about segmentation is incorrect?

a. Programs use two-component virtual addresses

b. Segmentation is designed to support a large number of segments at a time

c. The OS maintains a segment table shared by all processes

d. The task of designing a fully functional segmentation system is quite challenging

Answer: c

34. Major aspects of protection and security include the following except

a. Authentication b. Authorization c. Amortization d. Cryptography

Answer: c

35. Which is not a protection mechanism?

a. Cryptography b. Security policy c. Authentication d. Authorization

Answer: b

Test-V (True/False)

1) A DOS program is invoked by issuing a command.
2) The most important program in an operating system is the supervisor program.
3) Multiprogramming means that two or more programs can run simultaneously.
4) Time-sharing is effective because input/output speeds are so much faster than CPU speeds.
5) Resource allocation means that a given program has exclusive use of computer resources.
6) Background programs are usually batch programs
7) Virtual storage is a technique of memory management that appears to provide users with more memory space than is actually the case.
8) With the virtual storage technique, secondary storage is considered real storage.
9) Windows uses a graphical interface.
10) Shell is another name for page.
11) Utility programs avoid duplication of effort.
12) In a given memory system, all page frames are the same size.
13) Time-sharing is both event-driven and time-driven.
14) Virtual memory is another name for virtual storage.
15) Loading the operating system into memory is called booting.
16) An interrupt causes a program to stop temporarily.
17) Tine-sharing programs usually operate in the background of memory.
18) Paging divides a program into pieces of various sizes to fit in the available memory spaces.
19) Response time is the time it takes a program to run.
20) OLE lets a user link from one program to another.
21) Multiprocessing is simultaneous processing.
22) Multiprogramming is one approach to sharing the CPU.

23) All operating system programs must be in memory during the time an applications program is running.
24) A typical time-sharing application is processing payroll checks.
25) An operating system is a virtual machine.
26) An operating system is a resource manager.
27) Job control language (JCL) described the employment of computer programmers.
28) Batch operating systems streamlined usage of computer resources.
29) Multiprogramming makes both multi-user and multi-tasking systems possible.
30) Multitasking is when several users can send tasks to the same machine.
31) Preemptive multitasking is when users determine which task is using the processor.
32) In a LAN, a client is a person who is requesting services.
33) Access to the Internet typically requires access to a network server.
34) Booting a system is giving a recalcitrant machine a swift kick.
35) To enable a PC to run Unix instead of DOS, you must reprogram the ROM.
36) The scheduler responds to events--such as when a user clicks on a mouse.
37) A program counter is similar to the program address register, only every process has one.
38) Virtual memory is limited by the amount of RAM a machine has.
39) A dispatcher responds to interrupts by switching from one process to another.
40) Thrashing is when a frustrated user is shaking a keyboard or a mouse.
41) A spool can make more effective use of high quality printers and other resources.
42) Computer security has to do with keeping unauthorized users out of a computer room.
43) Protection and security are more important for networked than stand-alone computers.
44) Obscure words like 'ROM' or 'cipher' or 'gnostic' make good passwords.
45) A public-key encryption scheme is unsuitable for securing data over the Internet.
46) Viruses and other malicious software and spread via floppy diskettes or over networks.
47) An application program is a set of system software routines that sits between the operating system and the hardware.
48) The user communicates with the application program, the application program communicates with the operating system, and the operating system communicates with the hardware.
49) Most application programs access the hardware directly.
50) A service is a hardware link activated by the operating system.
51) Because different brands of computers often imply different hardware, application software written for one brand on computer cannot run on another.
52) The advantage of using layers of abstraction is that you can focus on the details of any given layer without losing sight of the other layers.
53) The idea behind viewing a system as a set of layers of abstraction is derived from an old architectural concept called layering.
54) The contents of a black box are unknown to the other black boxes, so each layer is functionally independent.
55) Two black boxes communicate with each other only through a shared interface or point of linkage.
56) Viewing each layer of abstraction as an independent black box is useful because it allows you to work with the layers one at a time.
57) The operating system's user interface, sometimes called the shell, provides a mechanism for the user and application programs to communicate directly with the hardware.
58) The operating system's file management function, sometimes called the file system, incorporates routines that allow the user or programmer to create, delete, modify, and manipulate files logically, by name.
59) The operating system's device management function is responsible for efficiently managing the system's memory and processor.

60) Processor (or process) management is concerned with efficiently managing the processor's time and main memory space.

61) Memory management is concerned with managing the system's memory resources as the computer runs, allocating space to applications as needed and ensuring that those applications do not interfere with each other.

62) Inter-computer communication is enabled by processor management services installed on both machines.

63) Because each layer is functionally independent, layers cannot be combined to form a system.

64) All operating systems are open source; in other words, they are based on open, published source code that can be modified and improved by anyone.

65) UNIX is an example of a proprietary operating system.

66) Some peripheral devices are linked directly to the processor.f

67) A buffer is temporary memory or storage used to adjust for the speed differential between the processor and memory.

68) On a mainframe computer, a channel handles device-independent functions and device-dependent functions are implemented through I/O control units or interface units.

69) The operating system's job is to manage the computer system's resources as efficiently as possible, but the precise definition of efficiency depends on the computing environment.

70) Throughput is typically defined as total execution time (for all programs) divided by total elapsed time, and is often expressed as a percentage.

71) Turnaround is typically defined as the elapsed time between job submission and job completion.

72) Response time is typically defined as the elapsed time between a request for the computer's attention and the computer's response.

73) Security is the probability that a system will perform as expected for a specified period of time.

74) Memory management is concerned with managing the computer's available pool of memory, allocating space to application routines and making sure that they do not interfere with each other.

75) Processor management is concerned with managing the processor's time.

76) Some routines, such as the ones that control physical I/O, directly support application programs as they run and thus must be transient.

77) Resident routines are stored on disk and read into memory only when needed.

78) The transient area is where resident operating system routines are loaded.

79) Generally, the more programs in memory, the greater the utilization of the processor.

80) The simplest approach to managing memory for multiple, concurrent programs, fixed-partition memory management, divides the available space into fixed-length partitions each of which holds one program.

81) Under dynamic memory management, the transient area is treated as a pool of unstructured free space and a program is loaded into a region allocated at load time.

82) With virtual memory management, little chunks of unused space spread throughout memory create a fragmentation problem.

83) With segmentation, programs are divided into independently addressed segments and stored in contiguous memory.

84) On a segmented system, the dynamically expanded address consists of two parts: a segment number and a displacement.t

85) With segmentation and paging, addresses are divided into a segment number and a displacement, but the segments are paged into memory.

86) If a program attempts to modify (or, sometimes, even to read) the contents of memory locations that do not belong to it, the operating system's memory protection routine intervenes and (usually) terminates the program.

87) Loading only currently active pages is the underlying principle behind modern overlay systems.

88) The external paging device is usually RAM.

89) Virtual memory is a model that simplifies address translation.

90) Swapping pages between the external paging device and real memory is done in response to a user request, so the user must be aware of the process.

91) When a virtual address points to a page that is already in real memory, a page fault is recognized and a page-in (or swap-in) operation begins.

92) Bringing pages into memory only when they are referenced is called demand paging.

93) Pre-paging involves predicting the demand for a new page and swapping it into memory before it is actually needed.

94) Pre-paging occurs when the system finds itself spending so much time swapping pages into and out from memory that little time is left for useful work.

95) Memory mapping is a technique for managing memory.

96) Traditionally, the key measure of multiprogramming effectiveness is response time.

97) The system operator manages the processor's time by determining which application program is executed next.

98) On many systems, a control block is created to hold a partition's key control flags, constants, and variables.

99) The operating system knows when input and output operations begin and end because these events are marked by interrupts.

100) Hardware processes an interrupt by saving key control information for the currently executing program and starting the interrupt handler routine.f

101) Interrupts can originate with either hardware or software.

102) Processor management is concerned with the internal priorities of programs already in memory.

103) As programs enter the system, they are placed on a queue by the scheduler.f

104) Once a program is in memory, the dispatcher uses its external priority to determine its right to access the processor.

105) Until the program is in memory, it has no internal priority. Once in memory, its external priority is no longer relevant.

106) The most important measure of effectiveness of a time-sharing system is response time.

107) During idle periods, the operating system's spooling module reads data from such slow devices as terminal keyboards and stores them on a high-speed medium such as disk. Later, when the program is loaded, its input data can be read from high-speed disk.

108) Fragmentation occurs when two (or more) programs each control a resource needed by the other.

109) A modern, well-designed operating system still has design flaws.

110) The OS is the part of the system software that manages the use of the hardware by application software that chooses to take advantage of its (the OS) services.

111) System software must be useful in most application domains.

112) Abstraction simplifies the way an application program controls the hardware, but can also limit flexibility to manipulate the hardware.

113) An abstraction cannot be simpler than the actual resource interface.

114) In a conventional, single-CPU computer system, multiple programs can execute simultaneously.

115) Multiple program executions each appear to have their own private computer—an abstract machine—on which to execute.

116) Well-designed multiprogramming can improve the performance of most processes.

117) Resource isolation is mandatory for the correct operation of most abstract machines.

118) All system software is implemented as trusted software.

119) A classic process represents the concept of a program in execution.

120) The combination of the OS functions and user mode instructions defines the abstract machine interface.

121) In a user space thread model, the interface is provided by a user space library, but the implementation is inside the OS kernel.

122) The hardware distinguishes between instructions that can be used without affecting the resource sharing model and those are a used to provide a robust sharing model.

123) Context switches can occur whenever the OS gets control of the processor.

124) The maximum number of simultaneous processes is usually static in an OS.

125) OS software is designed to favor performance over maintainability.

126) Most resources used by a thread are allocated to the associated process rather than the thread itself.

127) It is generally possible to predict the amount of time a thread will hold a resource.

128) A consumable resource is allocated but not released.

129) A multiprogramming OS allows more than one process at a time to be loaded into the primary memory, but limits processes to one thread each.

130) The cost of context switching is a modest factor in considering CPU multiplexing operations.

131) The yield operation and hardware interrupts are largely unrelated concepts.f

132) Infinite loops are a problem for voluntary sharing strategies.

133) An interrupt system can force periodic interrupts of some, but not all, processes.

134) The scheduler can have a dramatic effect of the performance of a multiprogrammed computer.

135) The interrupt timer device handler works closely with the process to coordinate actions.

136) Non-preemptive scheduling algorithms allow any process/thread to run to "completion" once it has been allocated the processor.

137) In preemptive scheduling algorithms, the highest-priority thread is allocated the CPU.

138) In nonpreemptive scheduling algorithms, it is common to ignore the cost of context switching.

139) The concurrent execution of two threads that access the same shared variable are not guaranteed to be determinate if not every execution is guaranteed to produce the same results.

140) A race condition exists when the outcome of computation depends on the relative times that two processes execute their corresponding critical sections.t

141) enter() and exit() library function calls can be used to solve the general critical section problem.

142) Deadlock can occur in any situation where multiple processes compete for resources.

143) The critical section problem is a special case of the mutual exclusion problem.t

144) A process can be created/destroyed in about the same amount of time as sophisticated synchronization operations.

145) Semaphores are only useful for solving critical section problems, and not other synchronization problems.

146) Semaphores eliminate the need for program correctness proofs.

147) Semaphores can be implemented without disabling interrupts by the use of advanced user mode programming techniques in modern languages.

148) Counting semaphores cannot be constructed from simple binary semaphores.

149) New synchronization methods such as AND, Events, and Monitors enable solutions to problems that cannot be solved solely with semaphores.

150) Nested semaphore operations can lead to deadlock.

151) Event semantics are constant across virtually all systems.

152) A major distinction between events and semaphores is that if no thread is waiting when a signal is raised, the result of signal() operation is not save and its occurrence has no effect.

153) A monitor forces a process to wait if another process is currently executing one of the monitor's member functions.

154) Monitor member function execution is treated like a critical section.

155) When a thread is running a monitor function, it may be interrupted by another thread within the same process.

156) Monitors prevent deadlock.

157) When threads are in different processes (with no shared memory), the OS must assist the threads in sharing information.

158) Placing mailboxes in user space is preferable to placing them in system space.

159) A thread should always be able to distinguish between being temporarily blocked and deadlocked.

160) Deadlock is no different in processes with multiple threads, than with single-threaded processes.

161) OS services are immune from deadlock.

162) Violating mutual exclusion as a means to implement deadlock prevention is not possible for every resource type in a conventional system.

163) The only wait to deal with the hold-and-wait condition in deadlock prevention is to require processes to release all currently held resources prior to requesting new ones.

164) An acyclic process-resource graph is evidence of a deadlock.

165) Interactive systems lend themselves to hold-and-wait prevention strategies that require processes to declare resource needs at process creation time.

166) The bakery algorithm is the classic avoidance strategy.

167) When a resource is preempted from a process, the process normally is destroyed.

168) A checkpoint/rollback strategy can be used to avoid deadlock.

169) At compile time, a static variable is allocated storage in the data segment and referenced by a relative address within the data segment.

170) At link time, the code and data segments for each relocatable object module are combined to form the object program.

171) In an absolute program, every reference to a data or a program entry point is resolved.

172) The malloc() function always gets additional address space from the OS.

173) In modern systems, the address space is an explicit part of the process abstraction.

174) Fixed-partition memory managers are suitable for a wide range of systems, including interactive systems.

175) Compaction requires that a program's address space be reduced.

176) Modern memory managers all use some form of variable partitioning.

177) Hardware dynamic relocation allows the OS freedom to choose (and later change) the memory location for an executable image.

178) In modern operating systems, only the OS can change relocation register contents.

179) In virtual memory systems, the memory manager is still the resource manager for the primary memory, although it allocates memory based on virtual memory policy instead of user program requests.

180) The 'trick' in virtual memory is that the process is unaware of how the virtual address space is bound to the primary memory.

181) Whenever a thread references a part of the virtual address space that is not currently loaded in the primary memory, the OS suspends execution and loads the missing information from another portion of primary memory.

182) A 'missing' memory element is detected before an instruction executes, to avoid the need to re-execute an instruction.

183) Program translation tools must be designed to support paging operation.

184) Paging is only feasible with special hardware support.

185) Pre-fetch policies are difficult to construct.

186) Demand-paging fetch policies are rare.

187) The dominant cost of paging is the I/O time for replacement.

188) The working set principle limits the amount of pages that a process can request.

189) In the earliest days of computing, designers saw little reason for decoupling CPU execution from I/O operation.

190) Device driver construction is a rigorous software design procedure.t

191) Application programmers prefer serial execution semantics for an individual thread, but will work with other computation models.

192) From the process's perspective, the abstract machin environment waits for the device to complete the I/O operation before continuing execution of the next statement.

193) If an individual thread is unable to take advantage of the overlap of CPU and I/O operations, the OS can overlap the CPU execution of one thread with the I/O operation of other threads.

194) Each I/O operation requires that the software and hardware coordinate their operations.

195) Multiprogramming may result in sporadic detection of I/O completion when using direct I/O with polling.

196) Polling is always superior from the viewpoint of an individual process.

197) Randomly accessed storage devices allow a driver to access blocks in an arbitrary order, without incurring an order-dependent performance penalty.

198) SSTF is a superior strategy because it successfully avoid starvation.

199) Protection mechanisms are almost always implemented as part of the OS rather than as user space software.

200) It is common to implement the protection mechanism as a trusted component of the system, and then allow the policy to be defined in an untrusted world.

201) Protection mechanisms introduce administrative overhead that can severely impact system performance.

202) Once an OS has been proven to be secure, the system is known to be secure.

203) Eavesdropping is probably less widely used than brute force attacks.

204) Brute force is especially effective when a computer is used to simulate a human connecting to a target computer.

Answers

1) T	2) T	3) F	4) F	5) F	6) T
7) T	8) F	9) T	10) F	11) T	12) T
13) F	14) T	15) T	16) T	17) F	18) F
19) F	20) T	21) T	22) T	23) F	24) F
25) T	26) T	27) F	28) T	29) T	30) F
31) F	32) F	33) T	34) F	35) F	36) F
37) T	38) F	39) T	40) F	41) T	42) T
43) F	44) F	45) F	46) T	47) F	48) T
49) F	50) F	51) F	52) T	53) T	54) T
55) T	56) T	57) F	58) T	59) F	60) F
61) T	62) F	63) F	64) F	65) F	66) F
67) F	68) T	69) T	70) T	71) T	72) T
73) F	74) T	75) T	76) F	77) F	78) F
79) T	80) T	81) T	82) F	83) F	84) T
85) F	86) T	87) F	88) F	89) T	90) F
91) F	92) F	93) T	94) F	95) F	96) F
97) F	98) T	99) T	100) F	101) T	102) T

103) F	104) F	105) T	106) T	107) T	108) F
109) T	110) F	111) F	112) T	113) F	114) F
115) T	116) F	117) T	118) F	119) T	120) T
121) F	122) T	123) T	124) T	125) T	126) T
127) F	128) T	129) F	130) F	131) F	132) T
133) F	134) T	135) F	136) T	137) T	138) T
139) T	140) T	141) F	142) T	143) T	144) F
145) F	146) F	147) T	148) F	149) F	150) T
151) F	152) T	153) T	154) T	155) F	156) F
157) T	158) F	159) F	160) F	161) F	162) T
163) F	164) F	165) F	166) F	167) T	168) F
169) T	170) F	171) T	172) F	173) T	174) F
175) F	176) T	177) T	178) T	179) T	180) T
181) F	182) F	183) F	184) T	185) T	186) F
187) T	188) T	189) T	190) T	191) F	192) T
193) T	194) T	195) T	196) F	197) T	198) T
199) T	200) T	201) T	202) F	203) F	204) T

13

DATA COMMUNICATION AND NETWORKS

Chapter Overview

- 13.1 Data Communication
  - 13.1.1 Characteristics of Data Communication System
  - 13.1.2 Basic Elements of Data Communication
- 13.2 Data Transmission Modes
- 13.3 Analog and Digital Data
- 13.4 Forms of Data Transmission
  - 13.4.1 Digital Data Transmission
  - 13.4.2 Analog Data Transmission
- 13.5 Forms of Signals
- 13.6 Modulation
  - 13.6.1 Types of Modulation
- 13.7 Serial and Parallel Transmission
- 13.8 Types of Data Transmission
  - 13.8.1 Asynchronous Transmission
  - 13.8.2 Synchronous Transmission
- 13.9 Computer Network
  - 13.9.1 Examples of Computer Network
  - 13.9.2 Network Criteria
  - 13.9.3 Type of Connection
  - 13.9.4 Benefits / Advantages of Computer Networks
  - 13.9.5 Disadvantages of Networks
- 13.10 Client and Server
  - 13.10.1 Dedicated Server
  - 13.10.2 Authentication Server
  - 13.10.3 File Server
- 13.11 Types of Computer Networks
  - 13.11.1 LAN (Local Area Network)
    - 13.11.1.1 Peer-to-Peer Network
    - 13.11.1.2 Client-Server Network
  - 13.11.2 WAN (Wide Area Network)
    - 13.11.2.1 Internet Peer-to-Peer
  - 13.11.3 MAN (Metropolitan-Area Network)
  - 13.11.4 Difference between LAN and WAN
- 13.12 Wireless Networks
  - 13.12.1 Categories of Wireless Networks
- 13.13 Network Topologies
  - 13.13.1 Bus Topology

- 13.13.2 Star Topology
- 13.13.3 Ring Topology
- 13.13.4 Tree Topology
- 13.13.5 Mesh Topology

13.14 Network Standards
- 13.14.1 Standards Organization

13.15 Protocol
- 13.15.1 Functions of Protocols
- 13.15.2 Key Elements of Protocol

13.16 Network Communication Technologies
- 13.16.1 Ethernet
  - 13.16.1.1 CSMA/CD
- 13.16.2 Token Ring
- 13.16.3 TCP/IP
- 13.16.4 Bluetooth
- 13.16.5 Wi-Fi
- 13.16.6 UWB
- 13.16.7 IrDA
- 13.16.8 WiMAX
- 13.16.9 WAP

13.17 Home Networks

13.18 Communications Software

13.19 Switched Networks
- 13.19.1 Circuit Switching
- 13.19.2 Packet Switching
- 13.19.3 Message Switching

13.20 Telephone Network
- 13.20.1 Dial-Up Lines
- 13.20.2 Dedicated Lines

13.21 Communication Devices
- 13.21.1 Modem
- 13.21.2 Network Interface Card
- 13.21.3 Network Bridge
- 13.21.4 Wireless Access Point
- 13.21.5 Hub
- 13.21.6 Network Switch
- 13.21.7 Managed Switches
- 13.21.8 Multilayer Switches
- 13.21.9 Routers
- 13.21.10 Gateway

13.22 Internetworks

13.23 Communications Channel
- 13.23.1 Physical Transmission Media
  - 13.23.1.1 Twisted Pair
  - 13.23.1.2 Coaxial Cable
  - 13.23.1.3 Fiber Optics
- 13.23.2 Wireless Transmission Media
  - 13.23.2.1 Microwave System
  - 13.23.2.2 Satellite Communication
  - 13.23.2.3 Broadcast Radio
  - 13.23.2.4 Cellular Communication
  - 13.23.2.5 Infrared

13.24 Media Connectors
13.25 Transmission Impairment
    13.25.1 Causes of Transmission Impairment
13.26 Fiber Distributed Data Interface (FDDI)
13.27 OSI Model
    13.27.1 Advantages of OSI Model
    13.27.2 Layered Architecture
    13.27.3 Peer-to-Peer Processes
    13.27.4 Interfaces between Layers
    13.27.5 Organization of the Layers
    13.27.6 Layers of OSI Model
13.28 TCP / IP Protocol Suite
    13.28.1 Physical and Data Link Layers
    13.28.2 Network Layer
    13.28.3 Transport Layer
    13.28.4 Application Layer
13.29 TCP/UDP Port Functions
13.30 IP Addressing
    13.30.1 IPv4 Addressing
    13.30.2 IPv6 Addressing
    13.30.3 IP Address Assignment
13.31 Protocols
    13.31.1 TCP
    13.31.2 UDP
    13.31.3 DHCP
    13.31.4 SMTP
    13.31.5 SNMP
    13.31.6 File Transfer Protocol (FTP)
    13.31.7 Secure File Transfer Protocol (SFTP)
    13.31.8 Trivial File Transfer Protocol (TFTP)
    13.31.9 POP3 and IMAP4
    13.31.10 Hypertext Transfer Protocol (HTTP)
    13.31.11 Hypertext Transfer Protocol Secure (HTTPS)
13.32 NAT
13.33 ICS
13.34 DNS
13.35 Static and Dynamic IP Addresses
13.36 Firewalls

Short Questions
Multiple Choices
Fill in the Blanks
True/False

13.1 Data Communication

Data communication is a process of transferring data electronically from one place to another. Data can be transferred by using different medium. Data communication is local if the communicating devices are in the same builiding. It is considered **remote** if the devices are at distant locations.

Data is collection of raw facts and figures. It may consist of text, graphics and sound etc. An electromagnetic or light wave used to transmit data from one place to another is called a **signal**. Propagation of signals across a communication medium is called **signaling**. Communication of data from one place to another place in the form of signals is called **transmission**.

13.1.1 Characteristics of Data Communication System

Data communication system consists of software and hardware. Effectiveness of a data communication system depends on three fundamental characteristics.

1. Delivery

The system must deliver data to the correct destination. Data must be received by the intended device or user.

2. Accuracy

The system must deliver data accurately. Data that have been altered in transmission and left uncorrected are unusable.

3. Timeliness

The system should deliver data within time. Data becomes useless if it is delivered late. In some cases, timely delivery means that data is delivered as it is produced. It includes the delivery of video, audio and voice data. It is known as **real-time transmission**.

4. Jitter

Jitter is the variation in the arrival time of the packet. It is the uneven delay in the delivery of audio or video packets. Suppose that audio packets are sent every 20ms. The audio quality will be uneven if some packets arrive in 20ms delay and some packets arrive in 30ms delay.

13.1.2 Basic Elements of Data Communication

The following components are required for successful communication to take place:

1. Message

The **message** is the data or information to be communicated. It may consist of text, number, picture, sound, video or a combination of these.

2. Sender

Sender is the device that sends the message. It is also called **source or transmitter**. The sender can be a computer, fax machine or mobile phone etc. The computer is usually used as sender in data communication systems.

3. Receiver

Receiver is the device that receives the message. It is also called **sink**. The receiver must be capable of accepting the message. The receiver can be a computer, printer, fax machine or mobile phone etc. A computer is usually used as receiver in data communication systems.

4. Transmission Medium

Transmission medium is the path through which the messages are transferred. It is used to carry messages from one place to another. It is also called **communication channel**. The transmission medium is a physical cable or wireless connection.

5. Protocol

A **protocol** is a set of rules that governs data communication. It represents an agreement between communication devices. Devices cannot communicate without protocol.

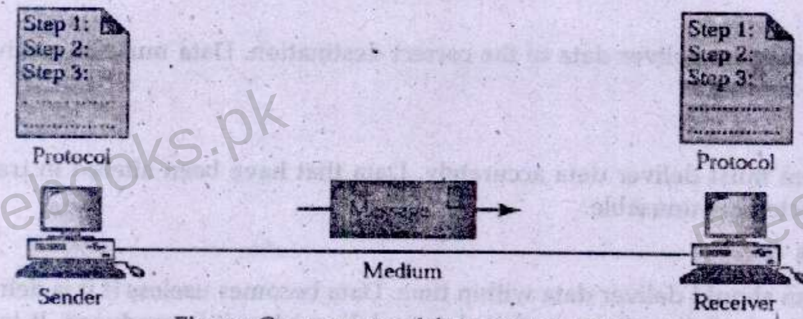
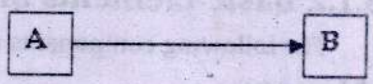


Figure: Components of data communication

13.2 Data Transmission Modes

The way in which data is transmitted from one place to another is called data transmission modes. There are three types of data transmission modes known as simplex mode, half duplex mode and full duplex mode.

1. Simplex Mode

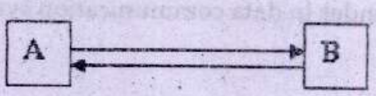


In **simplex mode**, data can flow only in one direction. It cannot be moved in both directions. It operates in a manner similar to a one-way street. The direction of flow never changes. A device with simplex mode can either send or receive data. It cannot perform both actions.

Example

An example is a traditional television broadcast. The signal is sent from the transmitter to TV antenna. There is no return signal.

2. Half-Duplex Traffic



In **half-duplex mode**, data can flow in both directions but not at the same time. It is transmitted one-way at one time. A device with half-duplex mode can send or receive data but not at the same time. That is why the speed of half-duplex mode is slow.

Example

Internet surfing is an example of half-duplex communication. The user issues a request for a web page. The webpage is downloaded and displayed before issuing another request.

3. Full-Duplex Mode

In **full-duplex mode**, data can travel in both directions simultaneously. Full duplex mode is a faster way of data transmission as compared to half duplex. Time is not wasted in changing the direction of data flow.

Example

A telephone is a full-duplex device. Both persons can talk at the same time. Another example of full-duplex communication is automobile traffic on a two-lane road. The traffic can move in both directions at the same time.

13.3 Analog and Digital Data

Data can be analog or digital. **Analog** means something continuous. An example of analog data is the human voice. When somebody speaks, a continuous wave is created in air. It can be received by a microphone and converted into an analog signal.

Digital means something discrete. An example of digital data is data stored in the memory of a computer in the form of 0s and 1s. It is usually converted to a digital signal when it is transferred from one position to another inside or outside the computer.

13.4 Forms of Data Transmission

The two forms of data transmission are digital and analog.

13.4.1 Digital Data Transmission

Digital signal is a sequence of voltage represented in binary form. The digital signals are in the form of electrical pulses of ON and OFF. These signals are in discrete form. Digital signals are faster and efficient. They provide low error rates. They also provide high transmission speed and high-quality voice transmission.

All data communication between the computers is in digital form. Computers understand and work only in digital form. The following figure represents a high voltage as a 1 and a low voltage as a 0.

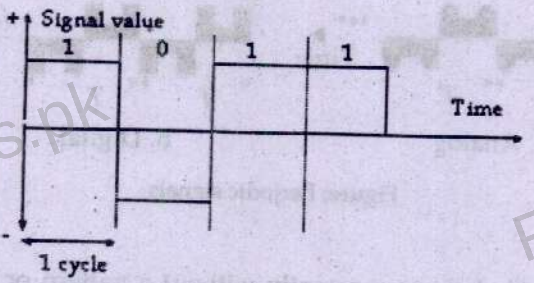


Figure: Digital Transmission

13.4.2 Analog Data Transmission

Analog signal is a continuous electrical signal in the form of a wave. The wave is known as carrier wave. Telephone line is most commonly used media for analog transmission of data. Light, sound, radio and microwave are also examples of analog signals.

Characteristics of Analog Signals

Two characteristics of an analog wave are as follows:

1. Frequency	The number of times a wave repeats during a specific time interval is known as frequency.
2. Amplitude	The height of wave within a given period of time is known as amplitude.

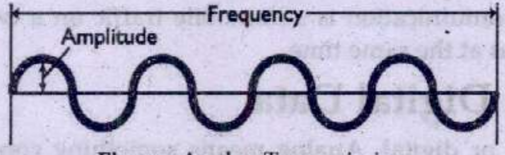


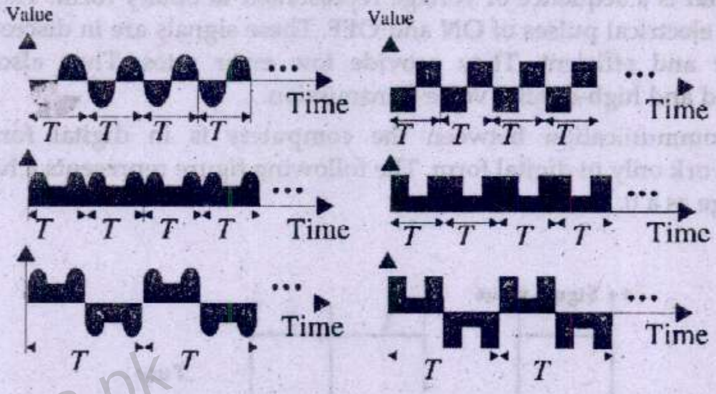
Figure: Analog Transmission

13.5 Forms of Signals

Both analog and digital signal can be in periodic or aperiodic form.

1. Periodic Signal

A signal is a periodic signal if it completes a pattern within a measurable time frame called period and repeats that pattern over identical subsequent periods. The completion of one full pattern is called a cycle. A period is defined as the amount of time required to complete one cycle. The duration of a period is represented by T and expressed in seconds.

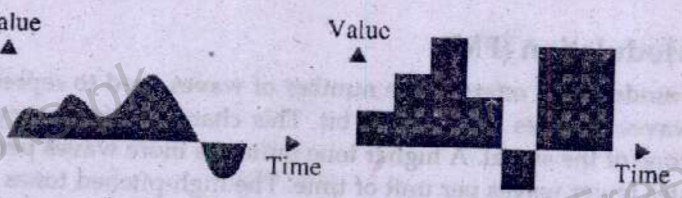


a. Analog b. Digital

Figure: Periodic signals

2. Aperiodic Signal

An **aperiodic signal** changes constantly without a pattern or cycle that repeats over time. It means that it has no repetitive pattern. An aperiodic signal can be decomposed into an infinite number of periodic signals. It is also called **non periodic signal**.


Value
Time
Value
Time

a. Analog b. Digital

Figure: Aperiodic signals

13.6 Modulation

The data on a network is transmitted through an electrical signal called **carrier signal.** The signal oscillates back and forth across a zero voltage line. The area above the line indicates a positive voltage and the area below the line indicates a negative voltage. That is why a carrier signal is also called **carrier wave or sine wave.**

A carrier signal or wave is an analog signal. Analog signal is a continuous wave whereas a digital signal consists of discrete elements. The analog signal requires some method to convert it into digital data. The change made to the signal is called **signal modulation or modulation.** The signal modulation is performed by a modem in many data communication networks.

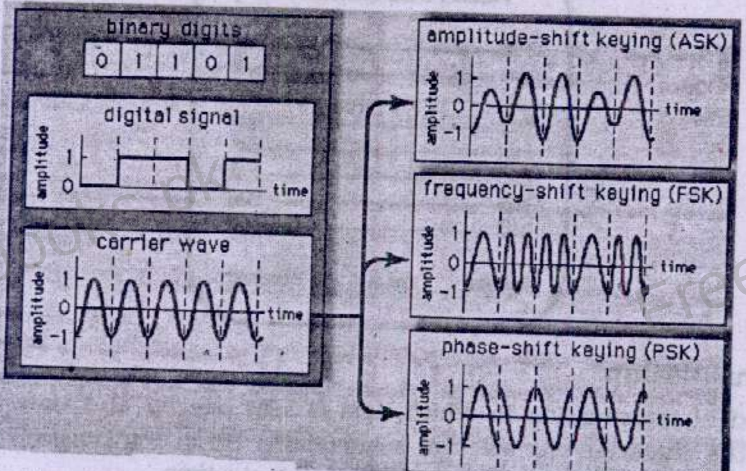
13.6.1 Types of Modulation

Three types of modulation used in a data communication network are as follows:

1. Amplitude Modulation (AM)

Amplitude Modulation refers to a change in the height of the carrier wave. The height of the signal is changed to indicate a 0 bit or 1 bit when data is sent on the carrier signal. The highest peaks of the signal represent 1 bits and the lowest peaks represent 0 bits. The high peaks in the sine wave produce louder sounds and the low peaks on the sine wave produce a quieter sound.

Amplitude modulation is often used for radio transmission specifically by AM radio stations. The AM signal is affected by noise and interference from external sources such as thunderstorm.


binary digits
0 1 1 0 1
digital signal
amplitude
1
0
time
carrier wave
amplitude
1
0
-1
time
amplitude-shift keying (ASK)
amplitude
1
0
-1
time
frequency-shift keying (FSK)
amplitude
1
0
-1
time
phase-shift keying (PSK)
amplitude
1
0
-1
time

2. Frequency Modulation (FM)

Frequency modulation refers to the number of waves used to represent a single cycle. The number of waves changes for a 0 or 1 bit. This change in frequency is indicated by a difference in the tone of the signal. A higher tone indicates more waves per unit of time and a lower tone indicates fewer waves per unit of time. The high-pitched tones indicate 1 bits and the low pitched tones indicate 0 bits in the above figure. Frequency modulation is used by FM radio stations to send radio transmission. It is also called **Frequency Shift Keying (FSK)**.

Frequency modulation is more resistant to external interference. FM radio signal is not interrupted by thunderstorm. The signals transferred using frequency modulation are less vulnerable to errors than signals sent by amplitude modulation.

3. Phase Modulation (PM)

Phase modulation is the third and most complicated type of signal modulation. A phase shift occurs to indicate a change in the types of bits being transmitted. The phase or direction of the wave changes when it goes from 0 to 1 bit or from 1 to 0 bit. The wave moves in a specific direction to indicate a 1 bit. The direction of the wave changes to 180 degree when a 0 bit is detected. The sine wave changes to the opposite direction each time a different bit is detected because the change is a 180 degree phase change. The direction changes but the amplitude and frequency of the signal remain the same. This type of phase modulation is called **Phase Shift Keying (PSK)**.

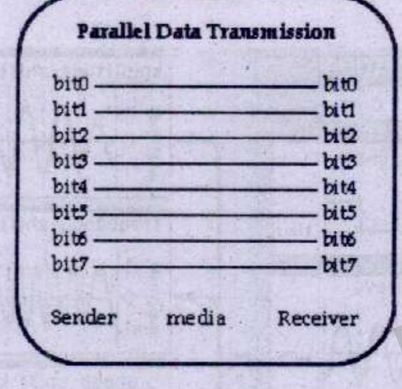
13.7 Serial and Parallel Transmission

The serial and parallel transmission are described as follows:

1. Parallel Transmission

A method of transmission in which groups of bits are sent at the same time over multiple wires is called **parallel transmission**. It is usually unidirectional. Each bit is transmitted over a separate line.

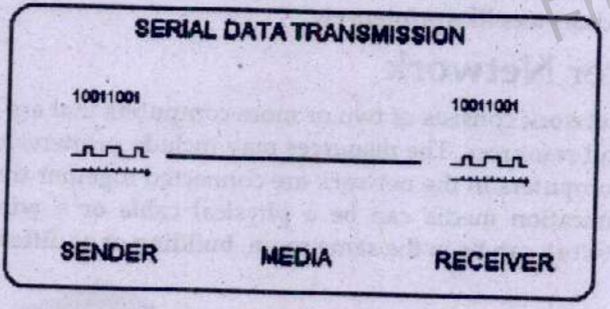
The internal transfer of data in a computer uses a parallel mode. The data transmission between computer and printer is done using parallel transmission. Parallel transmission is faster because all bits are sent at the same time.



2. Serial Transmission

A method of transmission in which data is sent one bit at a time is called **serial transmission**. The character bits are sent sequentially. Serial transmission is slower than parallel transmission as data is sent sequentially one bit at a time.

Telephone lines use this method of data transmission. Each individual bit of information travels along its own communication path.



13.8 Types of Data Transmission

Two types of data transmission are as follows:

13.8.1 Asynchronous Transmission

In **asynchronous transmission**, data is transmitted character by character. There are irregular gaps between characters in this transmission. It is cheaper to implement because data is not saved before it is sent.

It uses a special start signal. The signal is transmitted at the beginning of each message. The start signal is sent when the character is about to be transmitted.

A start bit has a value of 0. It is called **space state**. The value of 0 indicates that a character is about to be transferred. It alerts the receiver and it gets ready to receive the character. If start bit has a value 1, it indicates that the line is idle. It is also called **mark state**.

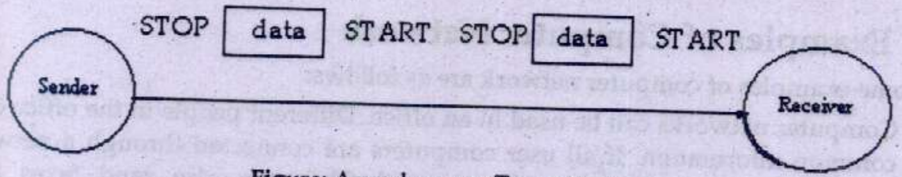


Figure: Asynchronous Transmission

13.8.2 Synchronous Transmission

In the **synchronous mode**, the saved data is transmitted block by block. Each block may consist of many characters. It uses a clock to control the timing of bits being sent. A large amount of information can be transmitted at a single time with this type of transmission.

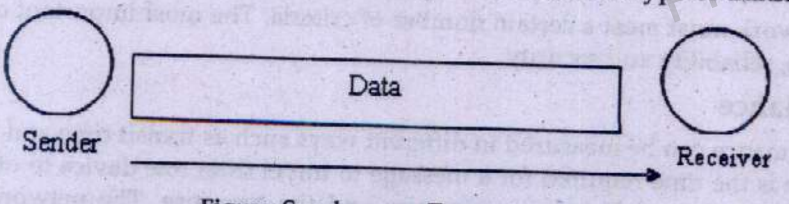


Figure: Synchronous Transmission

Synchronous transmission is much faster than asynchronous because there is no gap between characters. This transmission is suited for remote communication between a computer and related devices like printers etc.

13.9 Computer Network

A **computer network** consists of two or more computers that are connected together to share information and resources. The resources may include printers, hard disks, scanners or programs etc. The computers in the network are connected together through communication media. The communication media can be a physical cable or a wireless connection. The computers in the network can be in the same room, building or at different places.

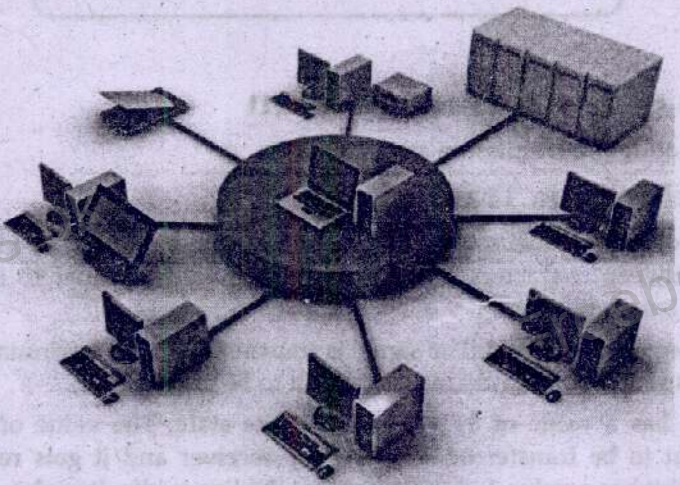


Figure: Computer Network

13.9.1 Examples of Computer Network

Some examples of computer network are as follows:

1. Computer networks can be used in an office. Different people in the office can access common information. If all user computers are connected through a network, they can share their files and exchange mail. They can also send faxes and print documents from any computer in the network.
2. Internet is also an example of a computer network in which millions of computers are connected through phone lines. People using this network can share information, files and talk with one another.

13.9.2 Network Criteria

A network must meet a certain number of criteria. The most important criteria include performance, reliability and security.

1. Performance

Performance can be measured in different ways such as transit time and response time. **Transit time** is the time required for a message to travel from one device to other. **Response time** is the time elapsed between an inquiry and the response. The network performnace depends on different factors such as number of users, type of transmission medium and the hardware and software being used.

The performance is evaluated by two metrics called **throughput and delay**. A higher throughput and less delays mean better performance. However, these can be contradictory. For example, the throughput may be increased by sending more data to the network. But it may result in increased delay due to congestion in the network.

2. Reliability

The **reliability** of the network is measured by different factors such as accuracy of data delivery. It also includes the frequency of failure and the time taken to recover from the failure etc.

3. Security

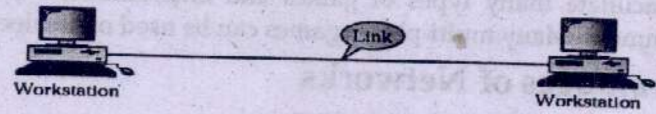
Security is the protection of data from unauthorized access, damage. It also includes the implementation of policies and procedures for data recovery if it is lost or damaged.

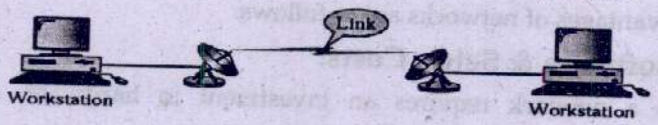
13.9.3 Type of Connection

A network consists of two or more devices connected through links. A **link** is a communication path that transfers data from one device to another. The communication can occur when two devices are connected to the same link at the same time. There are two possible types of connections between these devices.

1. Point-to-Point

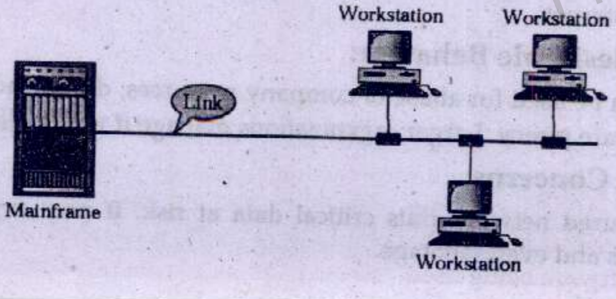
A **point-to-point connection** provides a dedicated link between two devices. The entire capacity of the link is reserved for data transmission between these two devices. Most point-to-point connections use physical wire to connect two devices. However, the wireless connection can also be used such as microwave or satellite link. An example of point-to-point connection is the link between a television and the remote control.





2. Multipoint

A **multipoint connection** is a connection in which more than two devices share a single link. Multipoint connection is also called **multidrop connection**. The capacity of the link in multipoint connection is shared among all devices spatially or temporally. The connection is called spatially shared if several devices can use the link at the same time. It is called timeshared connection if it can be used by one device at a time.



13.9.4 Benefits / Advantages of Computer Networks

Following are some important benefits of computer networks:

1. Information & Resource Sharing

Data and information can be shared among different users. Resources like printers, CD-ROM drives, hard disks and software can also be shared.

2. Money Saving

People can save a lot of money by using the same software over a network instead of purchasing separate copy of the software for each user.

3. Easy Communication

A person can communicate with others very easily using a large network like Internet.

4. Internet Access Sharing

Small computer networks allow multiple users to share a single Internet connection. Special hardware devices allocate the bandwidth of the connection to various individuals. An organization can purchase one high-speed connection instead of many slower ones.

5. Data Security and Management

A network can manage the company's critical data in a better way. Data can be centralized on shared servers instead of storing it on different computers. Everyone can find the data easily. It becomes easier for the administrators to take backup of data regularly. Security can be implemented easily

6. Entertainment

Networks facilitate many types of games and entertainment. Internet offers many sources of entertainment. Many multi-player games can be used over a local area network.

13.9.5 Disadvantages of Networks

Some disadvantages of networks are as follows:

1. Hardware, Software & Setup Costs:

Setting up a network requires an investment in hardware, software, planning, designing and implementing the network.

2. Hardware & Software Management Costs:

Managing a network is complicated. It requires training. A network manager usually needs to be employed. In a big organization, a network administrator is hired.

3. Undesirable Sharing:

Networking allows the sharing of undesirable data. Viruses are easily spread over networks and the Internet.

4. Illegal or Undesirable Behavior:

Networks can be used for abuse of company resources, downloading of illegal or illicit materials and software piracy. Larger organizations manage it with policies and monitoring.

5. Data Security Concerns:

A poorly-secured network puts critical data at risk. It may expose data to hackers, unauthorized access and even sabotage.

13.10 Client and Server

Client

A **client** is a computer in the network that is connected with a server to access different resources. The client computer sends request to the server for resources. The server computer provides the requested resource to the client computer. The client computer is less powerful than server computer.

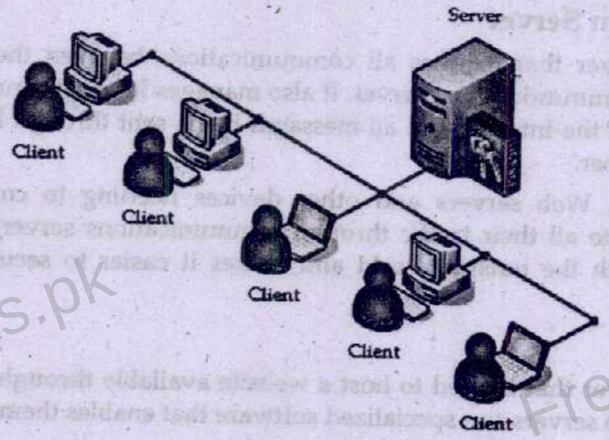


Figure: Client and Server

Server

A **server** is a computer that provides services to the computers and other devices connected to the network. Server computer is more powerful than other computers in the network. Different services provided by the server are as follows:

- Control access to the hardware, software and data
- Centralized storage for software, data and information
- Processing data
- Sharing software
- Managing network traffic

13.10.1 Dedicated Server

Dedicated servers are used to perform one specific function such as handling e-mail. Dedicated servers are used to perform the tasks that require a lot of time.

Different types of dedicated servers are as follows:

1. Print Server

A type of server that manages printing jobs is called **print server**. A network with a centralized printing service will have a **print server**. This server routes print jobs to the appropriate devices. It also prioritizes the printing according to order of request.

2. Application Server

A type of server that stores and distributes a set of application software to each system on the network is called **application server**. The user upgrades and installs new software once on application server rather than deploying the programs throughout the organization.

3. Database Server

A type of server that manages the database of an organization is called **database server**. It is dedicated to database storage, queries and retrievals etc.

4. Email Server

A type of server that handles a large volume of incoming, outgoing and internal email is called **email server**. It remains connected to the Internet. It may be located at any place in the world.

5. Communication Server

A type of server that handles all communications between the network and other networks is called **communications server**. It also manages Internet connectivity. All requests for information from the Internet and all messages being sent through Internet pass through communications server.

Email servers, Web servers and other devices needing to communicate with the Internet usually route all their traffic through communications server. It provides a single point of contact with the outside world and makes it easier to secure the network from hackers.

6. Web Server

A type of server that is used to host a website available through the Internet is called **web server**. The web servers run specialized software that enables them to host webpages.

13.10.2 Authentication Server

Authentication server keeps track of the users who log on to the network. It also keeps track of different services available to each user on the network. Authentication servers also act as overseers for the network. They manage and coordinate the services provided by any other dedicated servers located on the network.

13.10.3 File Server

File servers store and manage files for network users. On corporate networks, employees are provided with space on a file server to store files.

13.11 Types of Computer Networks

Computers networks are categorized according to:

- How they are organized physically.
- The way they are used.
- The distance over which they operate.

Three main types of computer networks are as follows:

- LAN - Local-Area Network
- WAN - Wide-Area Network
- MAN - Metropolitan-Area Network

13.11.1 LAN (Local Area Network)

LAN stands for **Local Area Network**. It is the most common type of network. It covers a small area. It usually connects the computers and other devices within one office or a building or group of buildings. LAN is often used to share resources such as printers, hard disks and programs.

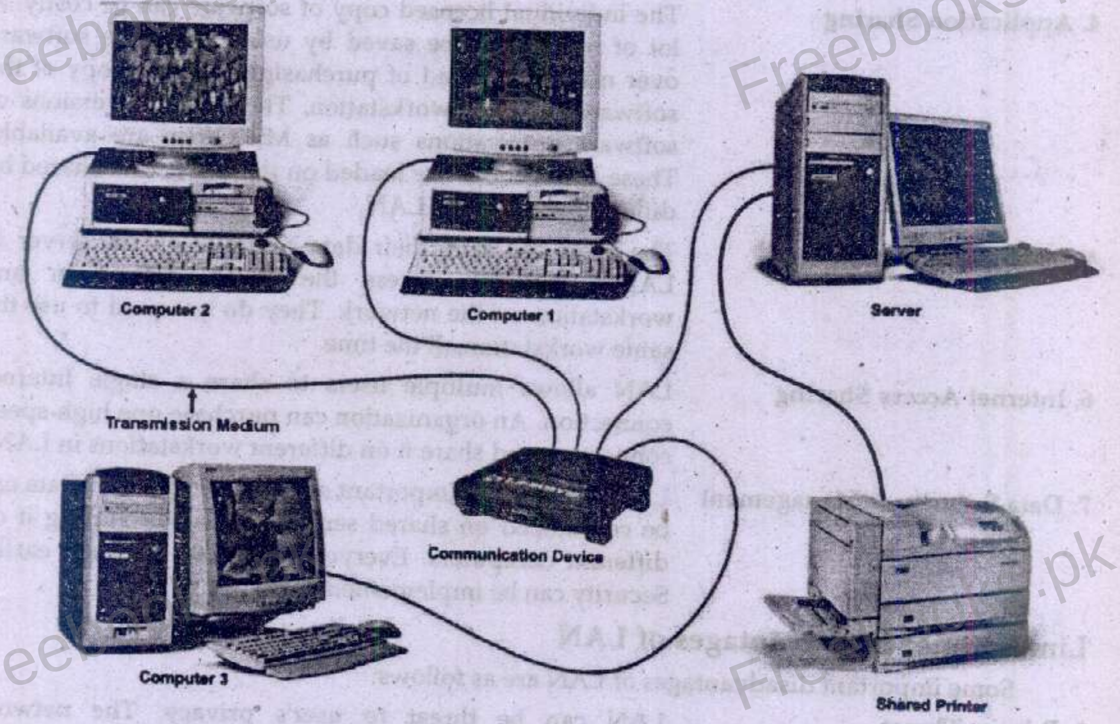


Figure: Local Area Network

LANs are capable of transmitting data at a very fast rate. LAN transmission speed is typically 10 Mbps to 1000 Mbps. It is much faster than data transmission over a telephone line. LAN can transmit data in a limited distance. There is also a limit on the number of computers that can be attached to the network.

Examples

- In a computer lab, there are 40 computers connected through LAN. The students can share software, files and data in the lab.
- In Internet club, many computers can be connected through LAN. These computers can share single connection of the Internet.

Benefits / Advantages of LAN

Some important advantages of LAN are as follows:

1. Resource Sharing	The resources like printers, CD-ROM drives, hard disks and software can be shared using LAN. This is cheaper than buying devices for each workstation in the network.
2. Communication	The users in LAN can easily communicate with each other. They can also transfer data easily and rapidly between different computers in the network.
3. Centralized IT Admin	LAN allows an easier and more efficient way to manage different computers from server. IT administrator can remotely troubleshoot the problems on computers across the network instead of doing it individually.

4. Application Sharing	The individual licensed copy of software can be costly. A lot of money can be saved by using the same software over network instead of purchasing separate copy of the software for each workstation. The network versions of software applications such as MS Office are available. These applications are loaded on the server and shared by different users in the LAN.
5. Centralized Data	The users can save their data centrally on the server in LAN. They can access the saved data from any workstation on the network. They do not need to use the same workstation all the time.
6. Internet Access Sharing	LAN allows multiple users to share a single Internet connection. An organization can purchase one high-speed connection and share it on different workstations in LAN.
7. Data Security & Management	LAN can manage important data in a better way. Data can be centralized on shared servers instead of storing it on different computers. Everyone can find the data easily. Security can be implemented easily.

Limitations / Disadvantages of LAN

Some important disadvantages of LAN are as follows:

1. Privacy Threat	LAN can be threat to user's privacy. The network administrator can access personal files of users. He can also monitor network and Internet activities of the user.
2. Expensive to Install	LAN generally saves money over time. However, it may require high initial costs of installation. The cables, network cards and software are expensive. The installation may also require the services of a technician.
3. Admin Time Required	The proper maintenance of LAN requires time and expertise. A network administrator may be required to manage network properly.
4. Data Security Concerns	LAN can put critical data at risk if it is not secured properly. The unauthorized users may access important data and can damage it.

13.11.1.1 Peer-to-Peer Network

Peer-to-peer LAN is a simple and inexpensive network. It normally connects less than ten computers. Each computer in this network can share hardware, data or information of any computer in the network. Each computer stores files on its own storage devices.

The network operating system and application software are installed on all computers. Any computer in the network can use the resources of any other computer in the network. Peer-to-peer networks are suitable for small businesses and home users. Windows provides peer-to-peer network utility.

Advantages

Some important advantages of peer-to-peer network are as follows:

- It is easy to setup.

- It is easy to maintain.
- It does not require an expensive server computer.
- It is suitable for small office of ten or less computers.

Disadvantages

The disadvantages of peer-to-peer network are as follows:

- Heavy use can slow down the network speed.
- It also provides less security of data because files are stored at different locations in the network.

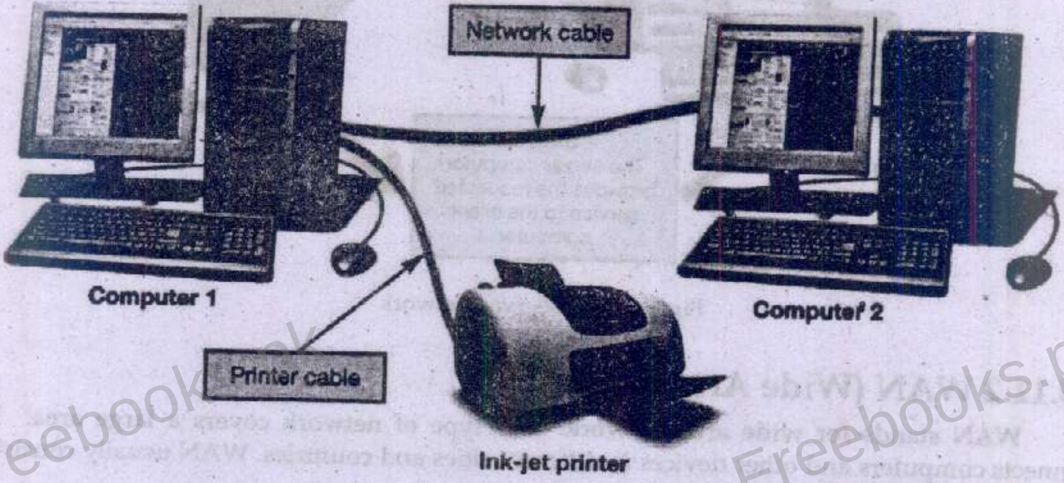


Figure: Peer-to-Peer Network

13.11.1.2 Client-Server Network

Client-server LAN is a network in which one or more computers work as servers and other computers work as clients. The **server computer provides services for the clients. Server** is also called **host computer.** It controls access to the hardware and software on the network and provides centralized storage area for programs, data and information.

Client computers request resources from the server. Server computer is more powerful than client computers and contains more memory.

Some servers are known as **dedicated servers.** A dedicated server is used to perform a specific task. For example, **file server** is used to store and manage files. **Print server** is used to manage printers and print jobs. **Network server** is used to manage network traffic.

Advantages

Some important advantages of client-server network are as follows:

- It reduces the volume of data traffic on the network.
- It also provides faster responses to the clients.
- It can use less powerful computers as clients because most of the processing is done by the server computer.

Disadvantages

- It is an expensive model because server computers are costly.
- The operations stop all over the network when server goes down.

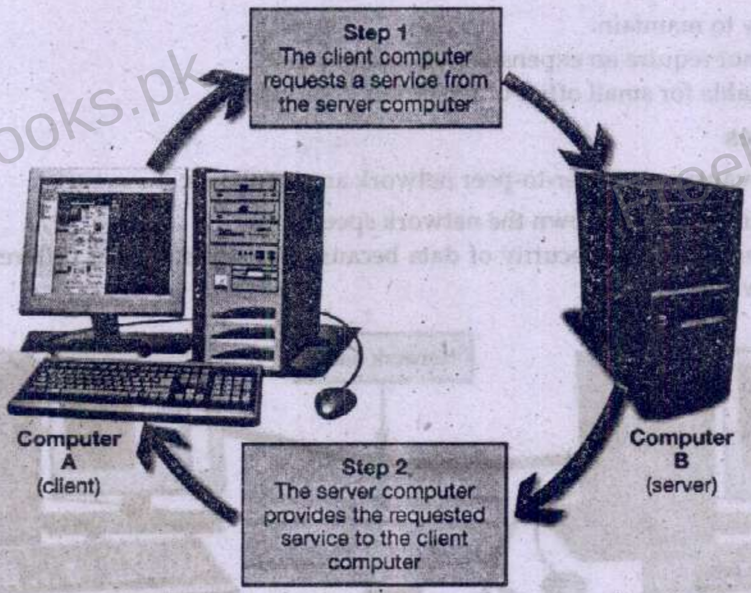

Step 1
The client computer requests a service from the server computer
Computer A (client)
Computer B (server)
Step 2
The server computer provides the requested service to the client computer

Figure: Client-Server Network

13.11.2 WAN (Wide Area Network)

WAN stands for wide area network. This type of network covers a large area. It connects computers and other devices in different cities and countries. WAN usually consists of several LANs connected together.

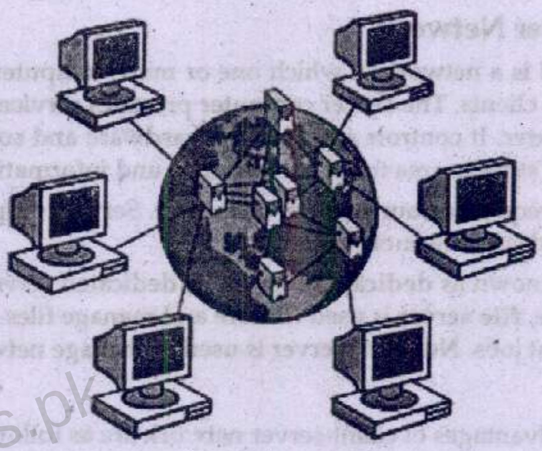


Figure: Wide Area Network

Computers in a WAN are often connected through telephone lines. They can also be connected through leased lines or satellites. WAN can reach the parts of the world that is not possible with LANs. WAN is expensive than LAN. WAN is not as fast as LAN. The transmission rate of WAN is typically 56 Kbps to 50 Mbps.

Examples

- The network connecting the ATMs of a bank located in different cities
- The network connecting NADRA offices in different cities of Pakistan
- Internet connects million of users all over the world to share information

Advantages of WAN

Some important advantages of WAN are as follows:

1. Communication Facility	A big company may exist at multiple locations in a country. The employees can communicate using WAN. It saves long distance phone calls. Video conferencing is another use of WAN where users can communicate with each other through their computer system.
2. Remote Data Entry	Remote data entry is possible in WAN. The user can sit at any location and enter, update and process data on any computer attached to WAN. For example, the user can access the data on a computer located at Karachi while sitting in Faisalabad through WAN.
3. Centralized Data	The centralized data storage is an important advantage of WAN. It means that data can be stored in single location even if the organization is spread over many cities. WAN can manage critical data in a better way by storing it on a centralized server. The centralized data can be shared among different users anywhere in the world.
4. Entertainment	WAN can facilitate many types of games and entertainment to the user. For example, Internet provides many sources of entertainment. Many multi-player games are available that can be played over WAN.

Disadvantages of WAN

Some important disadvantages of WAN are as follows:

- **Hardware, Software & Setup Costs:** Setting up a WAN requires an investment in hardware, software, planning, designing and implementing. The cost of devices used in WAN is very expensive.
- **Hardware & Software Management Costs:** Managing a WAN is complicated. It requires intensive training. A network manager usually needs to be employed.
- **Data Security Concerns:** A poorly-secured WAN puts critical data at risk. It may expose data to hackers, unauthorized access and even sabotage. Virus can spread quickly across the WAN if it enters the central backing store.
- **Failure of Server:** If a server fails, all computers connected with the server are affected.

13.11.2.1 Internet Peer-to-Peer

Internet peer-to-peer network is used by users to connect to each other's hard disks and exchange files directly. The users can copy files from each other's computers. Napster and Gnutella are two examples of networking software used in Internet peer-to-peer.

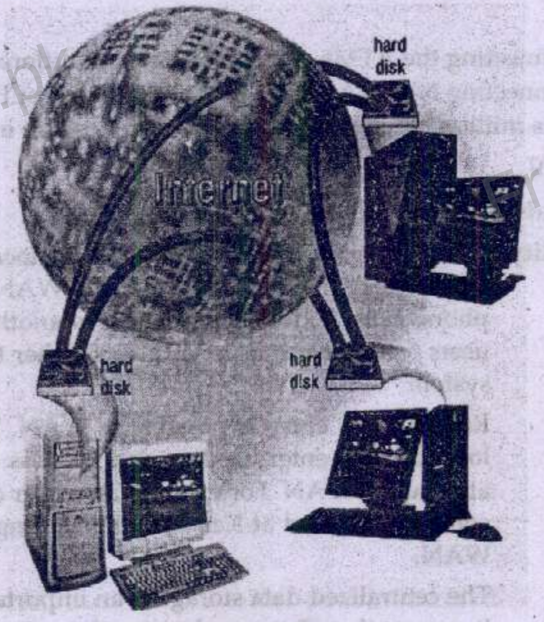

hard disk
Internet
hard disk
hard disk

Figure: Internet Peer-to-Peer

13.11.3 MAN (Metropolitan-Area Network)

MAN stands for Metropolitan Area Network. This type of network covers an area of a city. MAN is larger than LAN but smaller than WAN. It is usually used to connect two or more LANs in a city or town.

Examples

- The network connecting different branches of a company in same city
- The network connecting different campuses of a college in a city
- Cable TV network in a city

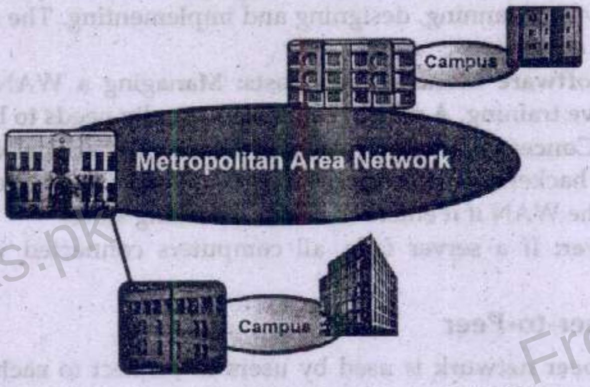

Campus
Metropolitan Area Network
Campus

Figure: Metropolitan Area Network (MAN)

Advantages of MAN

- MAN covers a larger area than LAN.
- It provides higher data speed than LAN.

Disadvantages of MAN

- It is more expensive than LAN.
- It is difficult to maintain as compared to LAN.

13.11.4 Difference between LAN and WAN

LAN	WAN
1. LAN is used to connect computers at one place.	1. WAN is used to connect computers anywhere in the world.
2. LAN covers limited area.	2. WAN can cover more distance.
3. Data transfer speed is very fast in LAN i.e. from 10 to 1000 Mbps.	3. Data transfer speed is slow in WAN i.e. from 56kbps to 50Mbps.
4. LAN is less costly.	4. WAN is expensive.
5. LAN is usually connected through wires.	5. WAN is usually connected through telephone lines.
6. The connection in a LAN is permanent using wires.	6. The connection in WAN is not permanent.
7. LAN is used for sharing files and hardware like printers, modem etc.	7. WAN is used to share only data and information like Email and file transfer
8. LAN has less possibility of data transmission error.	8. WAN has higher possibility of data transmission error.
9. The problems normally occur due to cable disturbance by the end user.	9. The problem normally occurs due to communication problems in medium.

13.12 Wireless Networks

Wireless network is a network set up by using radio signal frequency to communicate among computers and other network devices. It is also called **WiFi network or WLAN.** It is becoming popular because it is easy to setup and requires no cabling. The user can connect computers anywhere at home without the need for wires.

A **wireless network adapter** or **wireless router / access point** is used to setup wireless network. The wireless adapter connects a laptop or PC to a wireless LAN. A wireless router connects subnetworks or different independent WLANs together.

13.12.1 Categories of Wireless Networks

Different categories of wireless networks are as follows:

1. Fixed Wireless System A fixed wireless system provides a little or no mobility of the computing equipment used in wireless networks. The equipment remains at the same position. For example, a LAN can be setup using wireless technology to avoid cables. It will work as simple LAN but it does not require any cable.

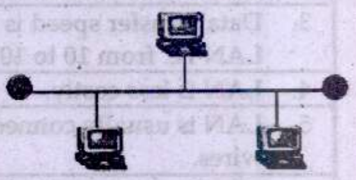
2. Mobile Wireless System A mobile wireless system provides mobility of computing equipment used in wireless networks. It also supports the mobility of users so mobile users can access information from anywhere and at any time. The user does not need to maintain fixed position in network. The typical equipments used in mobile wireless systems are smartphones, personal digital assistants and pagers with the Internet access.

13.13 Network Topologies

A network can be configured or arranged in different ways. The physical layout or arrangement of connected devices in a network is called **topology**. It is the shape of a network. Different network topologies are as follows:

- Bus Topology
- Ring Topology
- Star Topology
- Tree Topology
- Mesh Topology

13.13.1 Bus Topology



Bus topology is the simplest type of network. It supports a small number of computers. In bus topology, all computers or network nodes are connected to a common communication medium. This medium is often a central wire known as bus. The terminators are used at the end of a bus to absorb signals. A collision can occur in bus topology if two computers.

Working of Bus Topology

The sending computer sends the data and destination address through the bus. The data and address move from one computer to the other in the network. Each computer checks the address. If it matches with the address of a computer, the computer keeps the data. Otherwise the data moves to the next computer.

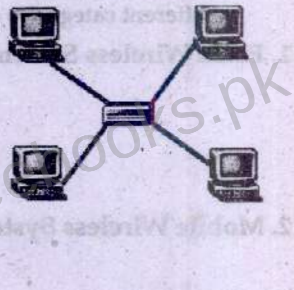
Advantages

1. It is simple and easy to use.
2. It requires small length of cable to connect computers.
3. It is less expensive.
4. It is easy to extend a bus. It allows more computers to join the network.
5. If one node fails, it does not affect the rest of the network.

Disadvantages

1. It is difficult to troubleshoot.
2. It only supports small number of computers.
3. If number of computers increases, network speed will slow down.

13.13.2 Star Topology



All computers in star topology are connected with the central device called hub. Star topology is mostly used in client-server networks.

Working of Star Network

The sending computer sends the data to the hub. The hub sends data to the receiving computer. Each computer in star network communicates with a central hub.

Advantages

1. It is easy to maintain and modify the network.
2. Adding or removing computers can be done without disturbing the network.
3. Finding the faults becomes very simple.
4. Single computer failure does not bring down the whole network.

5. It is more flexible among the remaining topology.

Disadvantages

1. The entire network breaks down if the central hub fails.
2. It requires a large length of cable to connect computers.
3. It is more expensive.

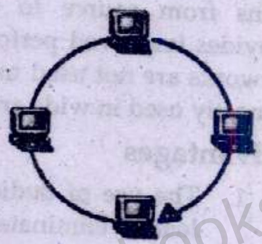
The Best LAN Topology

Star topology is the best LAN topology. Scalability and reliability of star topology make it the best topology than others. It is easy to remove or add a device in this topology. It is easier to troubleshoot than other topologies.

13.13.3 Ring Topology

In this topology, each computer is connected to the next computer with the last one connected to the first. Thus, a ring of computers is formed.

Working of Ring Network



Every computer is connected to the next computer in a ring. Each computer receives the message from the previous computer and transmits it to the next computer. The message flows in one direction. The message is passed around the ring until it reaches the correct destination computer.

Advantages

1. It is less expensive than star topology.
2. Every computer has equal access to the network.

Disadvantages

1. The failure of one computer in the ring can affect the whole network.
2. It is difficult to troubleshoot.
3. Adding or removing computers affects the whole network.

13.13.4 Tree Topology

A tree topology combines the characteristics of bus and star topologies. It consists of different groups of computers attached in star topology. The groups are then connected to a bus backbone cable. Tree topology is used for the expansion of an existing network.

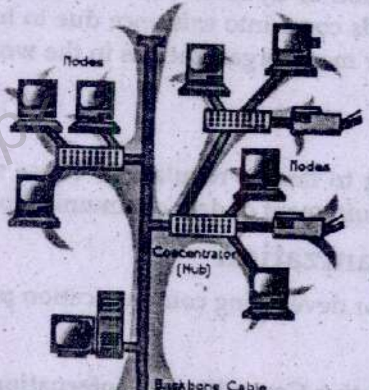


Figure: Metropolitan Area Network (MAN)

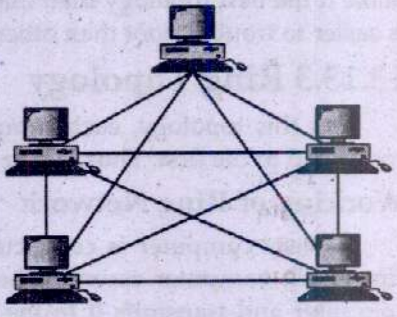
Advantages of a Tree Topology

1. It provides point-to-point wiring for individual segments.
2. It is supported by several hardware and software vendors.

Disadvantages of a Tree Topology

1. Overall length of each segment is limited by the type of cabling used.
2. If the backbone line breaks, the entire segment goes down.
3. It is more difficult to configure and wire than other topologies.

13.13.5 Mesh Topology



In a **mesh topology**, every device in the network is physically connected to every other device in the network. A message can be sent on different possible paths from source to destination. Mesh topology provides improved performance and reliability. Mesh networks are not used much in local area networks. It is mostly used in wide area networks.

Advantages

1. The use of dedicated link guarantees that each connection can carry its own data load. It eliminates the traffic problem.
2. If one link becomes unusable, it does not harm the entire system.
3. It is easy to troubleshoot.

Disadvantages

1. A full mesh network can be very expensive.
2. It is difficult to install and reconfigure.

13.14 Network Standards

The standards are the documents that contain technical and physical specifications about the network being designed. The networks can be reliable and efficient by following certain standards. The two types of network standards are as follows:

1. De Facto

De Facto means by tradition or by facts. These standards are developed without any formal planning. These standards come into existence due to historical developments. These standards are still being used by many organizations in the world. **SNA** is an example of De Facto standard.

2. De Jure

De Jure means according to law or regulation. These standards are developed with proper research to fulfill the requirement of data communication.

13.14.1 Standards Organization

The major organization for developing communication protocols and standards are:

i. CCITT

CCIT stands for **Consultative Committee on International Telegraph and Telephone.** It is United Nations agency. It is responsible for defining standards for telephone, telegraph and data communication.

The most popular standard defined by CCITT is X25. This standard is most common standard used in Wide Area Network (WAN). CCITT is now renamed as **International Telecommunication Union (ITU).**

ii. IEEE

IEEE stands for **Institute of Electrical and Electronics Engineers.** It also defines data communication protocols. The following communication protocols used in Local Area Network (LAN) are defined by IEEE:

- Ethernet (also known as IEEE 802.3)
- Token Bus (also known as IEEE 802.4)
- Token Ring (also known as IEEE 802.5)

iii. ISO

ISO stands for **International Standard Organization.** It is responsible for defining standards in various fields.

13.15 Protocol

A standard used by networks for communication between different devices connected to a network is called **protocol.** It represents an agreement between the communication devices. A protocol defines what is communicated, how it is communicated and when it is communicated. The device cannot communicate without a protocol.

13.15.1 Functions of Protocols

The main functions of protocols are as follows:

1. Data Sequencing

A process of breaking a long message into smaller blocks is called **data sequencing.** A long message is divided into smaller packets of same size. This technique reduces the amount of data that is retransmitted if an error is detected.

2. Data Routing

Data routing is the process of finding the most efficient path between source and destination before sending data. This technique increases the efficiency of data communication.

3. Data Flow

All computers are not equally efficient in speed. Data flow is a process that controls data transmission properly if the sender computer is faster than the receiver computer.

4. Error Control

Error detecting and recovering is an important function of communication protocol. It ensures that data is transmitted without error. It also solves problem if an error is detected.

13.15.2 Key Elements of Protocol

Three key elements of a protocol are as follows:

1. Syntax

Syntax refers to the structure or format of data. For example, a protocol may expect the first eight bits of data as the address of the sender, the second eight bits as the address of the receiver and rest of the stream as the message itself.

2. Semantics

Semantics refers to the meaning of each section of bits. It includes how should a particular pattern be interpreted and what action should be taken. For example, an address may identify the route to be taken or the final destination of the message.

3. Timing

Timing indicates when data should be sent and how fast it can be sent. For example, a sender may produce data at 100 Mbps when the receiver can process data at only 1 Mbps. The transmission will overhead the receiver and data will be lost..

13.16 Network Communication Technologies

There are different types of communications technologies that are as follows:

13.16.1 Ethernet

Ethernet is a LAN technology. It is based on bus topology but Ethernet network can be wired in star topology also. It is the most popular LAN because it is inexpensive and easy to install and manage.

Ethernet network normally uses cables to transmit data. Ehternet works properly for small to mid-sized networks. A recent standard of Ethernet called **Fast Ethernet.** It has data transfer rate of 100 Mbps which is 10 times faster than the old Ethernet. **Gigabit Ethernet** provides a higher transfer rate of 1 Gbps. 10 Gigabyte Ethernet provides the transfer rate of 6,375 Gbps.

13.16.1.1 CSMA/CD

Carrier Sense Multiple Access and Collision Detection (CSMA/CD) is the most common media access method for LANs. CSMA/CD is the MAC protocol used in Ethernet LANs. It essentially works in the following manner:

- Each node monitors the medium to see whether a message is being transmitted.
- Any node can begin a transmission if no message is detected.

The act of listening to the medium for a message is called **carrier sensing**. A carrier signal is present when a message is being transmitted. Several nodes can have messages to send. Each of them may detect a quiet medium and each may begin to transmit at one time. The ability of several nodes to access a medium that is not carrying a message is called **multiple access.**

A **collision** occurs if two or more nodes begin to transmit at the same time. Multiple simultaneous transmissions cause the messages to interfere with each other. The collisions must be detected and recovered. The messages are not transmitted successfully when a collision occurs. The sending nodes must send their messages again if a collision is detected. Another collision may occur if both nodes immediately attempt to retransmit their messages. Each node waits for a random interval before attempting to retransmit. It reduces the chance of another collision.

The **CSMA/CD** protocol is also called **listen-before-talk protocol**. It is a broadcast protocol. All workstations on the network listen to the medium and accept the message. Each message has a destination address. The workstation having the destination address can use the message. The use of a broadcast technique makes it easy to add and remove workstations from the network.

CSMA/CD is also known as **fair protocol**. It means that each node has equal access to the medium. In a pure CSMA/CD scheme, no one node has priority over another. However, the variations of this protocol exist to give one workstation priority over another to minimize the chance of collisions. One of these protocol variations divides time into transmission slots. The length of a slot is the time taken by a message to travel through the medium. The nodes on the network are synchronized. Each node can begin a transmission only at the beginning of its allocated time slot. This protocol has proven to be more efficient for the networks with many messages.

13.16.2 Token Ring

Token ring is also a LAN technology. It allows network devices to access the network by passing a special signal called token. Token is like a ticket. A device can transmit data over the network only if it has a token. Only one token is available in one network. That is why no collision can occur. But the data transmission rate is slow. When a computer wants to send a message, it:

1. Gets the token
2. Puts the data in the token
3. Adds the address of receiving computer

Token ring is based on ring topology but can also be used in star topology. The token passes from computer to computer. The computer whose address matches with the address stored in the token gets the data. It then returns the message to the sending computer to indicate that the data has been received.

13.16.3 TCP/IP

TCP/IP stands for **transmission control protocol/Internet protocol**. It is a technology that is used to manage the transmission of data by breaking it into packets. It is commonly used in Internet transmissions.

The data is divided into small pieces called **packets** when a computer sends data over the Internet. Each packet contains the data and the destination, sender and the sequence information used to reassemble the data at the destination. These packets travel via devices called **routers**. This process is known as **packet switching**.

13.16.4 Bluetooth

Bluetooth is a network standard and protocol. It defines the method for transmission of data between two Bluetooth devices using short-range radio waves. The devices can transfer data at a rate of up to 3 Mbps using Bluetooth. The devices must be within a distance of 30 feet. However, the distance can be extended using additional equipment.

A Bluetooth device uses a small chip to communicate with other Bluetooth devices. Many Bluetooth-enabled devices are available like desktop computers, notebook computers, smart phones, keyboards and printers etc. A Bluetooth wireless port adapter can be used to convert an existing USB port into Bluetooth port. Most of the current operating systems have built-in support for Bluetooth.

13.16.5 Wi-Fi

Wi-Fi is a popular networking technology that uses radio waves to provide high-speed wireless connections. It identifies any network based on **802.11** series of network standards. These standards specify how two wireless devices communicate with each other using air.

Wi-Fi is also called **wireless Ethernet** because it uses similar technique as Ethernet standard to specify the wireless network configuration. That is why Wi-Fi network can easily be integrated with Ethernet networks. Most computers and smart phones are Wi-Fi enabled.

An example of Wi-Fi network is the hot spots. A **hotspot** can be used by a mobile user to connect to the Internet using Wi-Fi enabled computers and devices. Wi-Fi can also be used to connect computers wirelessly at home and small business. The distance between different computers or devices can be up to 300 feet in open areas and about 100 feet in closed areas.

Standard	Transfer Rate	Description
802.11g	Up to 54 Mbps	It is the most commonly used standard. It is compatible with 802.11b.
802.11n	Up to 540 Mbps	It is approved recently and provides better speed and range.
802.11r	Up to 540 Mbps	It is an amendment to 802.11. It will improve VoIP and Internet-based technology
802.15	Up to 50 Mbps	It is used with Bluetooth technology. Its range is up to 32 feet.
802.16	Up to 70 Mbps	It is used to provide high-speed wireless Internet access over long distances of over 30 miles.
802.20	Up to 80 Mbps	It can be used in the vehicles moving at the speed of up to 250 km/hr.

Difference between Bluetooth and WiFi

The difference between Bluetooth and WiFi is as follows:

	Bluetooth	WiFi
Hardware requirements	• Bluetooth adapter for all devices connecting with each other	• A wireless adapter is for all devices in the network. • A wireless router or wireless access point
Developed in	1994	1991
Specification	Bluetooth SIG	IEEE 802.11
Ease of use	• Simple to use • Can connect up to 7 devices at a time • Easy to switch between devices	• Hardware and software configuration required
Primary devices	Mobile phone, mouse, keyboard	Notebooks, desktop computers and servers
Range	30 feet	300-500 feet
Security	More secure than WiFi	Security risks as other networks
Power consumption	Low	High

13.16.6 UWB

UWB stands for **ultra-wideband**. It is a network standard that specifies how two UWB devices communicate with each other at high speeds using short-range radio waves. The data transfer rate of UWB is 115Mbps at a distance of 30 feet. The data transfer rate increases to 480 Mbps if the distance is about 6 feet. UWB can transmit signals through walls and other obstacles. It can be used to transfer large files like video, audio and graphics. For example, it can be used to transfer video from digital video camera and download video to portable media player etc.

13.16.7 IrDA

IrDA is a network standard that is used to transmit data wirelessly via infrared light waves. It can be used by the devices to transfer data at the speed from 115 Kbps to 4 Mbps. Infrared requires a line-of-sight transmission. The sending and receiving devices must be in line with each other. There should be no obstacle between these devices.

13.16.8 WiMAX

WiMAX stands for **Worldwide Interoperability for Microwave Access**. It is a network standard developed by **IEEE** and is also known as **802.16**. It specifies how wireless devices communicate in a wide area over the air. The computers and devices with WiMAX capability can communicate using radio waves via WiMAX tower. A WiMAX tower can cover a radius of up to 30 miles. It connects to the Internet or another WiMAX tower.

WiMAX standard provides wireless broadband Internet access at reasonable cost. It can connect mobile users to the Internet via hot sports. Many mobile devices have built-in WiMAX capability such as smart phones.

Two types of WiMAX specifications include fixed wireless and mobile wireless.

Fixed Wireless WiMAX	It allows a user to access the Internet using desktop computer from a permanent location. It provides data transfer rate up to 35 Mbps.
Mobile Wireless WiMAX	It allows user to access WiMAX network with mobile computers and mobile devices. It provides data transfer rate up to 12 Mbps.

13.16.9 WAP

WAP stands for **Wireless Application Protocol**. It is used in mobile devices to access the Internet and its services such as the Web and email. It uses client/server network. The wireless devices contain the client software that connects to the server of the Internet server provider. The mobile devices that support WAP are called **WAP-enabled devices.**

13.17 Home Networks

A network of multiple computer at home are called **home network**. Home network allows different users to share hardware and software resources.

Some examples of home networks are as follows:

1. Ethernet Network

In Ethernet networks, computers are connected via cable. It is difficult to use as it may involve running cable through walls, ceilings and floors.

2. HomePLC Network

HomePLC (Powerline cable) network is also called **Powerline LAN**. It uses the lines that used in electricity. It requires no additional wiring. The original power line networks had a maximum data transfer rate of 14 Mbps. The new standards provide for data transfer rates approaching 200 Mbps.

3. Phoneline Network

It is an inexpensive and easy to install network. It uses telephone lines. It provides a maximum data transfer rate of 10 Mbps.

4. HomeRF Network

HomeRF (radio frequency) network uses radio waves to transmit data. It sends signals through the air. Current wireless networks in the United States are based on the **802.11 standard.** Each node on a wireless network requires a **wireless network adapter**.

Some vendors also provide an **intelligent home network**. This network provides the features such as lighting control, thermostat adjustment and security system etc.

13.18 Communications Software

Communication software consists of programs used to establish a connection to other computer or network. It manages transmission of data, instructions and information. Computers need compatible communications software to communicate properly with.

Some communications software are used to create a connection to another computer on the Internet. Some software are used to access the Internet using an ISP. Some software support file transfer protocol. **FTP (File Transfer Protocol)** is a standard used to upload and download files to and from a Web server over the Internet. This server is called **FTP server**.

13.19 Switched Networks

Switched network consists of interconnected nodes. Data is transferred from source to destination through these nodes. Data is switched from node to node so it is called switched network. Different types of switched networks are as follows:

13.19.1 Circuit Switching

Circuit switching creates a dedicated path between two nodes. The entire circuit must be available to transfer data. Data is sent on circuit-switching in three phases.

- The first phase establishes a circuit. This connection is required before data transfer.
- The second phase transfers the data.
- The third phase disconnects the circuit.

Telephone network is an example of circuit switching. First of all, a connection is established between two telephones. Secondly, two persons can talk with each other. Finally, the connection is terminated. Circuit switching is not very efficient. The entire circuit is dedicated for the duration of the connection and no other user can use it. It can also delay data transfer while the connection is being established.

13.19.2 Packet Switching

Packet switching network divides messages into fixed or variable sized packets and sends them to the destination. Packet switching is more efficient and less expensive than circuit switching. It is used by most of the modern message switching networks.

Network performance is increased in packet switching due to reduced size of packet. It uses three methods to send packets:

Datagram Services	It treats each packet independently. The packets contain sender's address, destination address and a sequence number in order to rebuild the message. The packet sent to the same receiver may use different paths and the receiver may receive them in wrong sequence. The receiver should be able to arrange the packets in proper order. A packet is called **datagram** in this method.
Switched Virtual Circuit	It establishes a logical connection between two nodes before sending the packet. The connection is established by a call setup request from the user. This connection is allocated for the duration of the session. The packets contain a code to identify virtual circuit used to reach the destination. The path for reaching the destination is determined before sending the packet. Each node in the circuit knows where to send the packet. Virtual circuit is different from dedicated path. Virtual circuit means that the circuit is determined before data transmission.
Permanent Virtual Circuit	It is used when two nodes require a continuous connection. It allocates a circuit permanently between two nodes and no call setup is required.

Difference between Circuit Switching and Packet Switching

Circuit switching is the switching method that is used in telephone networks. It sets up a temporary circuit between source and destination. The resources are reserved in the network to meet the service.

In packet switching, a message is divided into a series of segments or packets. Each packet contains the destination address and control instructions along with the message data. Data transfer on the Internet is based on packet switching. In this case, the Internet routers use the address information in each packet to send it on an efficient path to its destination. Different packets of the same message may follow different paths. The message is reassembled when all packets of the message are received at destination. The communications protocol such as TCP/IP can request a retransmission of missing packet if some packets are lost.

13.19.3 Message Switching

Message switching does not require a dedicated circuit. The message can be sent even if the receiver is not available. The message is stored until the receiver connects to the network. The messages may vary in length. The process of accepting message, storing it and passing it to the receiving node is called **store and forward**.

Store and forward process requires a computer at each node. Each computer accepts the message, stores it in a buffer, determines its destination and decides whether it should be sent to the network or not. The sending computer adds the address of destination node to the message. The processing at each node may cause some delay.

Email is an example of message switching. A user send email and it passes through several computers before reaching the receiver.

Difference between Message Switching & Circuit Switching Networks

Difference between message switching and circuit switching networks is as follows:

Message-switching network	Circuit-switching network
Good in efficiency	Poor line efficiency
Sender and receiver not on line simultaneous	Sender and receiver must be available simultaneous
Messages accepted on busy network	Messages blocked when busy
One message to several receivers	One message to one receiver
Can set priorities	No priorities
Error recovery per message	Little error recovery
Speed and code conversion	Difficult to do speed and code conversion
Stores messages for inoperative terminals	If receiver inoperative, message not sent

13.20 Telephone Network

Public switched telephone network (PSTN) is the worldwide telephone system that manages voice-oriented telephone calls. This network uses digital technology except the link from company to home that is often analog. It is also used in computer communications. Data, instructions, and information can be transferred over the telephone network using dial-up lines or dedicated lines.

13.20.1 Dial-Up Lines

Dial-up line is a temporary connection that uses analog telephone line for communications. A modem at the sending end is used to dial the number of a modem at the receiving end. When the modem at the receiving end answers the call, a connection is established and data can be transferred. If there is any problem in any of two modems, the communication stops.

Dial-up line is inexpensive as it costs a local call. It can be used to connect two computers located at any location. The quality of dial-up line is not very high.

13.20.2 Dedicated Lines

Dedicated line is a connection that is always established between two communications devices. The quality of dedicated line is better than dial-up line. It is used to connect two distant locations. These lines can be digital or analog.

The speed with which a line carries data and information is called **transfer rate**. The rate is expressed as **bits per second (bps)**. The transfer rate is from thousands of bits per second called **kilobits per second (Kbps)** to millions of bits per second called **megabits per second (Mbps)** and billions of bits per second called **gigabits per second (Gbps)**.

Different types of digital dedicated lines are as follows:

1. ISDN Lines

ISDN line provides faster transfer rates than dial-up telephone lines for small business and home users. ISDN stands for **Integrated Services Digital Network**. It is a set of standards for digital transmission of data over standard copper telephone lines. One telephone line can carry three or more signals at one time using the same line. This technique is known as **multiplexing**.

ISDN requires ISDN modem at both sides. ISDN modem is different from dial-up modem. It also requires special ISDN telephone for voice communications. The voice conversation and video conference become clearer.

2. DSL

DSL stands for **digital subscriber line**. It uses the standard copper telephone for fast speed transmission. A special network card of DSL modem is required to connect to DSL. DSL is easier to install than ISDN and is also faster.

A type of DSL is known as **ADSL**. It stands for **asymmetric digital subscriber line**. It provides faster transfer rates when receiving data than sending data. The rate of receiving data is called **downstream rate** and the rate of sending data is called **upstream rate**.

3. Cable Television Lines

Cable television (CATV) lines is not a type of standard telephone line. It is very popular dedicated line used to connect to the Internet.

4. T-carrier Lines

T-carrier line is one of many digital lines that carry multiple signals over a single communications line. These lines use multiplexing to share the telephone line. They provide fast data transfer rates. These lines are expensive and only medium or large companies can afford them.

T1 line is the most popular T-carrier line. It is used to connect to the Internet backbone. A **T-1 line** can support 24 simultaneous voice or data channels and achieve a maximum throughput of 1.544 Mbps.

T-2 lines are composed of four T-1 lines and deliver a throughput of 6.3 Mbps. **T-3** lines are a bundle of 28 T-1 lines and deliver a throughput of 44 Mbps. **T-4 lines** contain 168 T-1 lines and provide 274 Mbps of throughput.

5. Asynchronous Transfer Mode (ATM)

ATM is a service to carry voice, data, video and multimedia at very high speed. It is used by Telephone networks, the Internet and other networks with heavy traffic.

Type of Line	Transfer rate
Dial-up	Up to 56 kbps
ISDN	Up to 1.54 Mbps
DSL	128 kbps to 8 Mbps
Cable TV	128 kbps to 50 Mbps
T1	1.544 Mbps
T2	6.3 Mbps
T3	44 Mbps
ATM	155 Mbps to 622 Mbps and can reach 8 Gbps

13.21 Communication Devices

The hardware used to transmit data, instructions and information between a sending and receiving device is called **communication device**. A communication device at sending end sends data to a communication channel. A communication device at receiving end receives data from communication channel. Sometimes, communication device may also convert the data from digital to analog and analog to digital signals.

13.21.1 Modem

Modem stands for **modulator/demodulator**. It is a commonly used communication device. Modem sends and receives data from one computer to another on the Internet through telephone lines. The sending and receiving computers both must have modems.

Working of Dialup Modem

Computer stores information in the form of digital signals. However, the information transmitted over the telephone lines is in the form of analog signals. The modem receives data from computer in digital form and converts it into analog form. This process is called **modulation**. It sends analog signals to other computer using telephone lines.

The modem on receiving computer receives data in analog form. It converts the analog data back into digital form. This process is called **demodulation**.

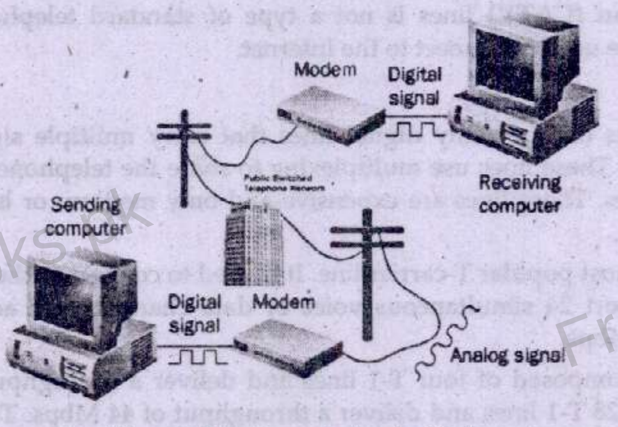

Modem
Digital signal
Public Switched Telephone Network
Receiving computer
Sending computer
Digital signal
Modem
Analog signal

Figure: Data transmission by Modem

Types of Modem

Different types of dial-up modems are as follows:

1. External Modem

External modem is attached to the system unit as an external device through telephone line. It is connected to the telephone wall jack by another cable. It requires external power supply. It is easy to setup. External modem is expensive.

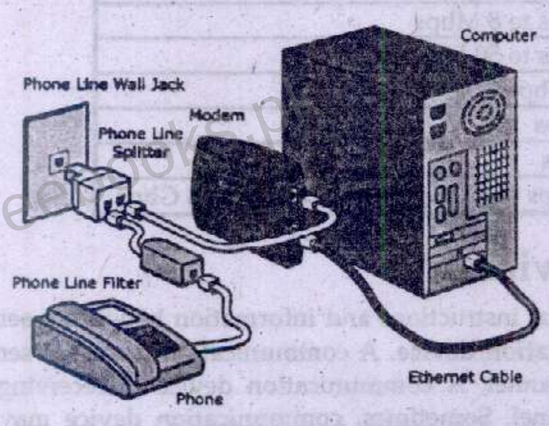

Computer
Phone Line Wall Jack
Modem
Phone Line Splitter
Phone Line Filter
Ethernet Cable
Phone

Figure: External Modem

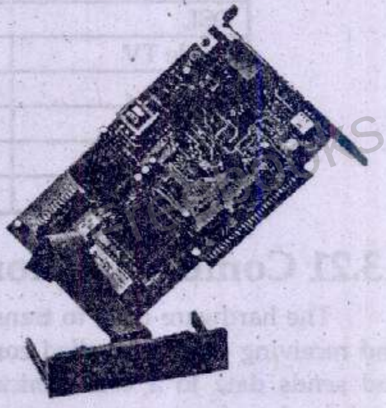
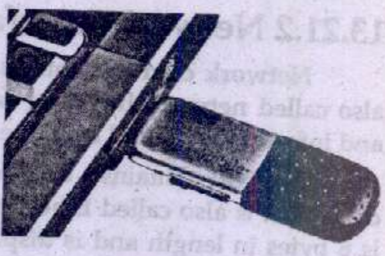


Figure: Internal Modem

2. Internal Modem

Internal modem is a circuit board that is inserted into an expansion slot on the motherboard. Internal modem cannot be moved from one computer to another easily. It is difficult to setup than other types of modem. It is less expensive than external modem.

3. Wireless Modem



Wireless modem transmits the data signals through the air instead of cable. It is also known as **radio-frequency modem**. It is designed to work with cellular technology and wireless local area networks. Most handheld computers, smart phones and other mobile devices use the wireless modem to connect to the Internet wirelessly. It can also be used with notebook and desktop computers.

4. ISDN Modems

ISDN modem allows digital transmission of data over telephone lines. It is also known as **ISDN adapter**. It uses the same twisted-pair wiring as dial-up modems but it can transmit data faster. It supports data transfer rates upto 1.54 Mbps.

5. DSL Modem

DSL modem allows faster transmission over the standard telephone lines. It is faster than ISDN. There are two types of DSL:

1. ADSL	It stands for **Asymmetric Digital Subscriber Line**. It uses faster transmission downstream than upstream. It is frequently used for faster internet services. It provides the upload speed of up to 640 Kbps and download speed of up to 8.1 Mbps.
2. SDSL	It stands for **Symmetric Digital Subscriber Line**. It uses the same transmission rate for both directions. It transfers data at 3 Mbps approximately in both directions.

DSL can only be used within three miles of a telephone switching station. Its speed degrades as the distance increases.

6. Cable Modem

Cable modem is also known as broadband modem. Cable modem is a stand-alone device connected with a cable to USB port. It sends and receives data over the cable television network. It provides faster Internet speed than dial-up modem, DSL modem and ISDN.

Figure: DSL Modem

Figure: Cable Modem

7. Satellite Modem

Satellite modem is used to send and receive data using satellite technology. It is commonly used for high-speed Internet service. Many satellite Internet services use satellite technology for downstream data only. It Satellite modem is a little slower than DSL and cable transmissions.

13.21.2 Network Interface Card

Network card is a communication device used to connect a device to a network. It is also called **network interface card (NIC). It manages the transmission of data, instructions** and information to and from the device.

The NIC contains a unique address called **MAC address**. MAC stands for **media access control**. It is also called **Ethernet, physical, hardware** or **adapter address**. The MAC address is 6 bytes in length and is displayed in 12 hexadecimal digits. The first 6 digits are used to indicate the vendor of the network interface card and is called **organizationally unique identifier (OUI)**. The last 6 numbers form a unique value for each NIC assigned by vendor. MAC address provides a way to distinguish one NIC from any other NIC.

Network cards are available in different styles. A network card for desktop computer is an adapter card. It has a port to connect the cable. A network card for mobile computer is in the form of USB network adapter, ExpressCard module, PC Card or memory card.

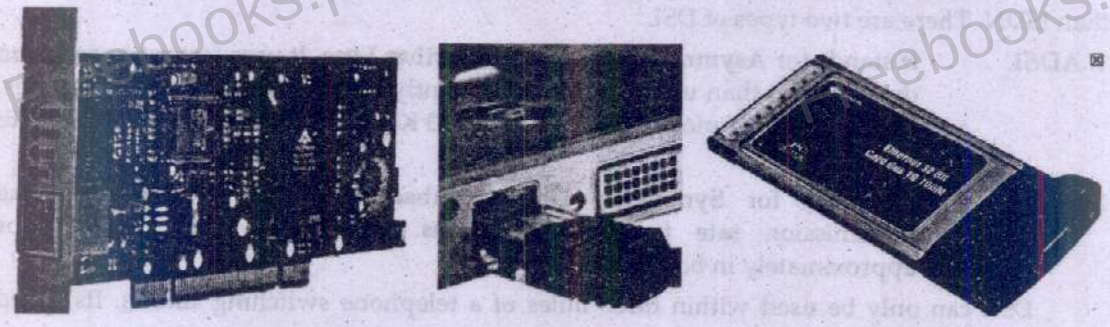


Figure: Network card

A wireless network card is used to transmit data wirelessly. It usually has an antenna that can be positioned properly for the best signals. An amplifier can be used for increasing the signal strength. Some network cards support wired as well as wireless networks.

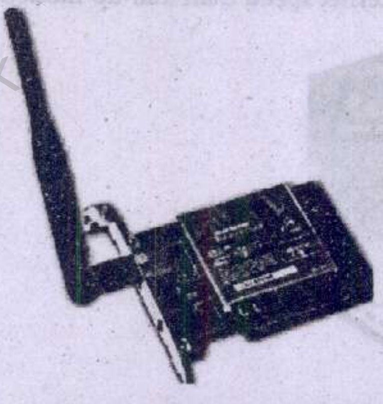


Figure: Wireless Network Card

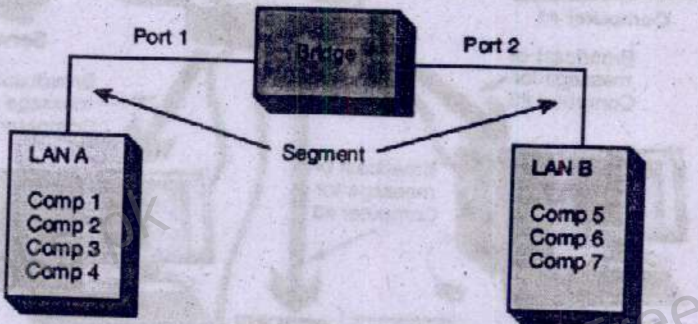
13.21.3 Network Bridge

A **network bridge** can be used in computer networks to interconnect two LANs and to separate network segments. A **segment** is a section of a network separated by bridges, switches and routers. The bridge is a layer 2 device in the OSI model. It uses MAC address information for making decisions to forward data packets. It only forwards the data that needs to be sent across the bridge to the adjacent network segment. This makes it possible to segment the network data traffic.

An example of using a bridge to segment two Ethernet LANs is shown in the figure. It shows that LAN A connects to port 1 of the bridge and LAN B connects to port 2. It creates two segments. There are four computers in LAN A and three computers in LAN B. Bridges monitor all data traffic in each LAN segments connected to its ports. A **port** is an input/output connection on a networking device. The bridges use the MAC addresses to build a bridging table of MAC addresses and port locations for hosts connected to the bridge ports. An example of bridging table is as follows. It shows the stored MAC address and the port where the address was obtained.

MAC Address	Port
00-40-96-25-85-BB	1
00-40-96-25-8E-BC	1
00-60-97-61-78-5B	2
00-C0-4F-27-20-C7	2

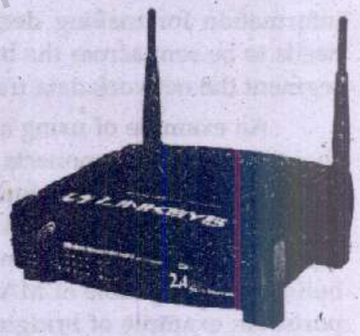
The source MAC address is stored in the bridge table as soon as a host transmits a data packet on LAN. Suppose that computer 1 in LAN A sends a message to computer 2. The bridge will store the MAC addresses of both computers and record that both computers are connected to port 1. If computers 5 or 6 are placing data packets on the network, the source MAC addresses for 5 and 6 are stored in bridge table and it is recorded that these computers connect to port 2 on the bridge. The MAC addresses for computers 3 and 4 will not be added to the bridging table until each transmits a data packet.



The bridge monitors the data on its ports to check for an association between the destination MAC address of the Ethernet frames to any of the hosts connected to its ports. An association indicates that the destination MAC address for a host is connected to one of the ports on the bridge. If an association is found, the data is forwarded to that port. Suppose that computer 1 sends a message to computer 5. The bridge detects an association between the destination MAC address for computer 5 and port 2. The bridge then forwards the data from computer 1 to computer 5 in LAN B through port 2.

The bridge forwards data packets only when there is an association. This capability is used to isolate data traffic in each segment. Suppose that computer 1 and computer 2 in LAN A generate a lot of data traffic. The computers in LAN B will not see any of the data traffic as long as there is not an association between the destination MAC addresses of the Ethernet packets and any of the hosts in LAN B.

13.21.4 Wireless Access Point



A **wireless access point** is a central communication device. It allows the computer and other devices to transfer data wirelessly among themselves or to a wired network.

Wireless access points have high quality antennas for optimal signals. Some manufacturers suggest to place the wireless access point at the highest possible location for the best signal.

13.21.5 Hub

A **hub** also called **concentrator or multistation access unit (MAU)**. It provides a central point for cables in a network. Hubs also transmit signals and have multiple ports to which devices are connected.

Hub is a physical layer device. It connects the computers to a network. It broadcasts all messages to every computer on the network but only the intended recipient computer takes the message. The computers that are not the recipients of the message ignore the message. A hub can handle only one-way information traffic at a time.

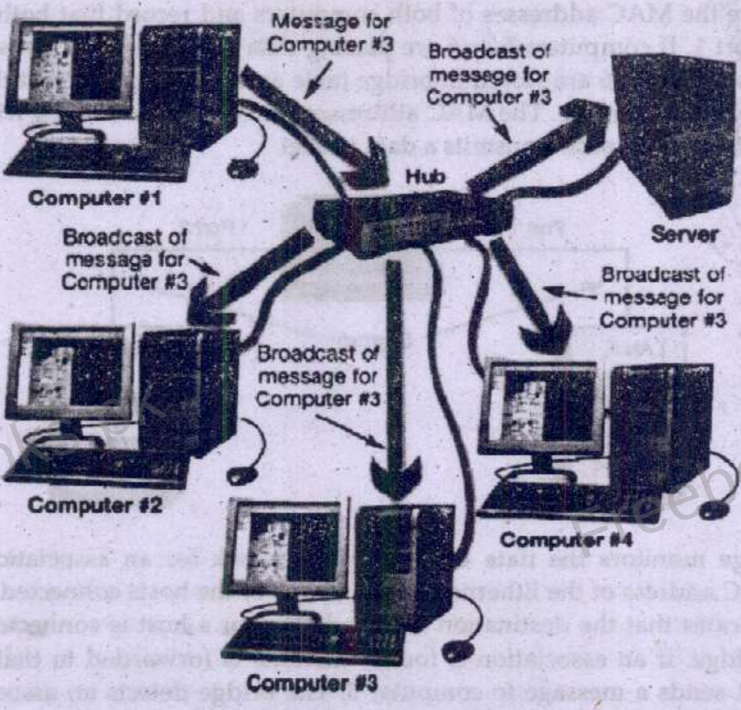
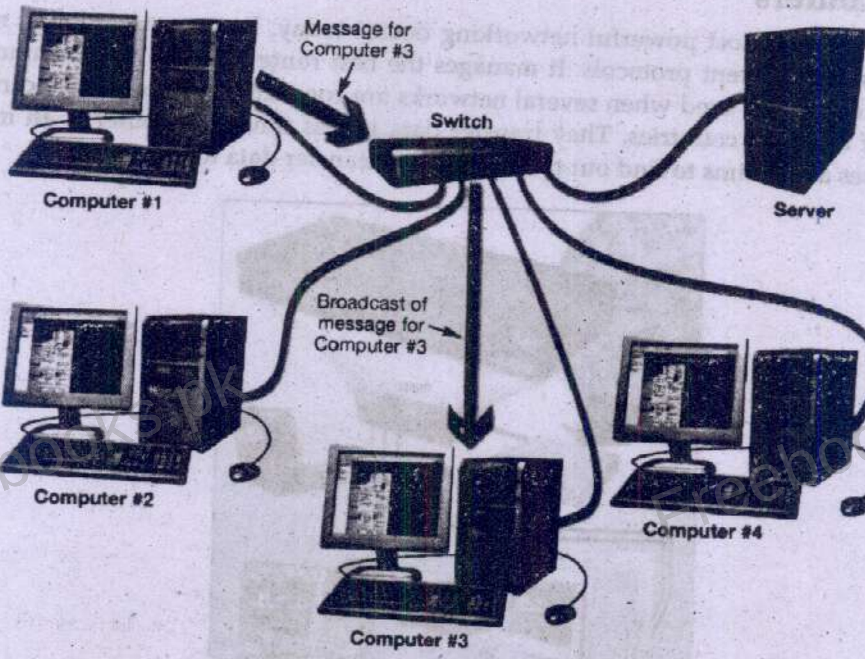

Message for Computer #3
Broadcast of message for Computer #3
Hub
Computer #1
Server
Broadcast of message for Computer #3
Broadcast of message for Computer #3
Broadcast of message for Computer #3
Computer #2
Computer #4
Computer #3

Figure: Hub in a Network

13.21.6 Network Switch

Network switch is more intelligent device than hub. The switch improves the efficiency of the data transfer in the network. It receives a message from any device connected to it and then transmits the message only to that device for which the message was meant. It can also handle multiple communications channels at the same time.



The switch operates at layer 2 of the OSI model. It uses MAC address for making decisions to forward data packets. The switch monitors data traffic on its ports and collects MAC address information like bridge to build a table of MAC addresses for the devices connected to its ports. The switch has multiple ports similar to the hub and can switch in a data connection from any port to any other port similar to the bridge. This is why the switch is sometimes called a multiport bridge. The switch minimizes traffic congestion and isolates data traffic in the LAN.

13.21.7 Managed Switches

A **managed switch** is a network switch that allows a network administrator to monitor, configure and manage different network features. The access to the management features for the switch is password protected.

13.21.8 Multilayer Switches

The **multilayer switches** can be used with multiple layers. These switches improve the performance of computer networks. An example is layer 3 switch. Layer 3 switch still works at layer 2 but additionally works at the layer 3 of the OSI model. It uses IP addressing for making decisions to route a data packet in the best direction. The major difference is that the packet switching in basic routers is handled by a programmed microprocessor. **The layer 3 switch uses application-specific integrated circuits (ASICs) hardware to handle the packet switching.**

The advantage of using hardware to handle the packet switching is a significant reduction in processing time. The processing time of layer 3 switches can be as fast as the input data rate. This is called wire speed routing where the data packets are processed as fast as they are arriving. Multilayer switches can also work at the upper layers of the OSI model. An example is a layer 4 switch that processes data packets at the transport layer.

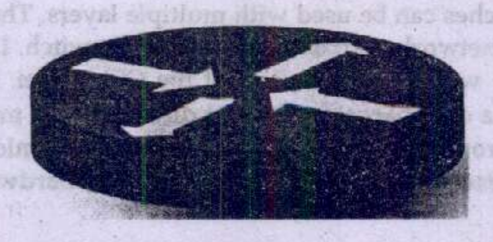
13.21.9 Routers

Router is the most powerful networking device today. It connects multiple networks using similar or different protocols. It manages the best route between two communication networks. Routers are used when several networks are connected together. They can connect networks of different countries. They transfer data in less time. The router is an intelligent device. It uses algorithms to find out the best path to transfer data to a network.

D-Link
router
cable or
DSL modem

The router is a layer 3 device in OSI model. It means that it uses the network address to make routing decisions to forward data packets. It is configured to know how to transfer data packets entering or exiting the LAN. This differs from the bridge and the layer 2 switch. The bridge and layer 2 switch use Ethernet address for making decisions to forward data packets. They only know how to forward data to hosts which are physically connected to their ports.

Routers have multiple port connections for connecting to LANs. A router must have a minimum of three ports. The most common symbol used to represent a router in a network is shown in the figure. The arrows pointing in and out indicate that data enters and exits the routers through multiple ports. The router ports are bidirectional and data can enter and exit the same port. The router ports are also called the **router interface**. It is the physical connection where the router connects to the network.



Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

13.21.10 Gateway

Gateway is a device that connects two or more networks with different types of protocols. It is an intelligent device. It can convert data according to the protocol. A gateway can be used to connect a personal computer network to mainframe computer network.

13.22 Internetworks

A collection or two or more networks is called **internetwork or network of networks.** It may consist of similar or different types of networks. These networks are connected with one another by using special devices. Most common devices are switches, routers, gateways and bridges.

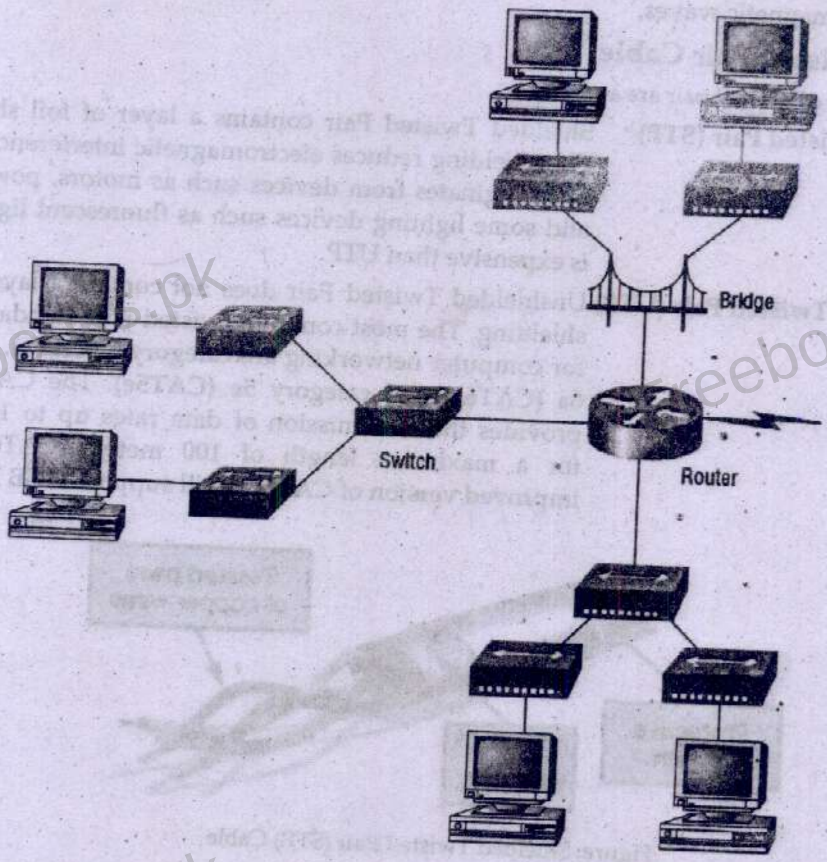


Figure: Network of networks

13.23 Communications Channel

The path through which data is transmitted from one place to another is called **communications channel.** It is also known as communication media or transmission media.

The amount of data that can be transferred through a communication medium in a unit of time is called **bandwidth.** The bandwidth of digital signal is measured in **bits per second** or **Bytes per second.** The bandwidth of analog signals is measured in **cycles/Seconds** or **Hertz.** There are different types of transmission media.

13.23.1 Physical Transmission Media

In **bounded media**, communication devices are directly connected with each other by using some physical media like wires. Some examples of bounded media are wire pair, coaxial cable and fiber optics.

13.23.1.1 Twisted Pair

Twisted pair is the most commonly used physical transmission medium. It is used in local area network to connect computers and other devices.

Twisted pair consists of a pair of copper wires. The pair of wires is covered by a plastic insulation and it is twisted together. Twisting of wires protects them from interference by external electromagnetic waves.

Types of Twisted Pair Cable

Two types of twisted pair are as follows:

1. **Shielded Twisted Pair (STP)** Shielded Twisted Pair contains a layer of foil shielding. The shielding reduces electromagnetic interference (EMI). EMI originates from devices such as motors, power lines and some lighting devices such as fluorescent lights. STP is expensive than UTP.
2. **Unshielded Twisted Pair (UTP)** Unshielded Twisted Pair does not contain a layer of foil shielding. The most common twisted-pair standards used for computer networking are **category 6 (CAT6), category 6a (CAT6a) and category 5e (CAT5e)**. The CAT6 cable provides the transmission of data rates up to 1000Mbps for a maximum length of 100 meters. CAT6a is an improved version of CAT6. It will support 10GB Ethernet.

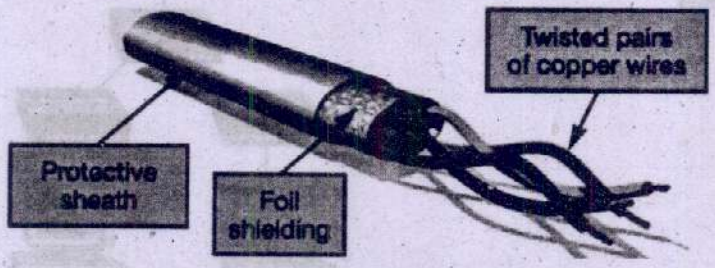


Figure: Shielded Twisted Pair (STP) Cable

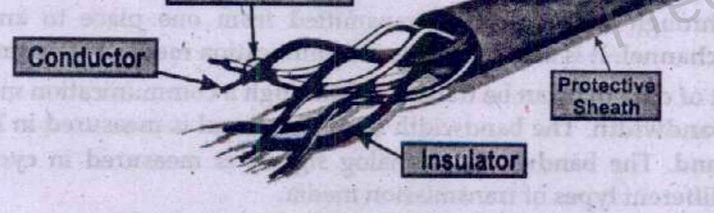


Figure: Unshielded Twisted Pair (UTP) Cable

Characteristics of Twisted Pair

Different characteristics of twisted pair are as follows:

- It is an inexpensive transmission medium.
- It is easy to install.
- It can transfer data to a short distance.

13.23.1.2 Coaxial Cable

Coaxial cable consists of copper wire covered by an insulating material. The insulated copper wire is covered by copper mesh. The mesh protects the data signals from interference by external electromagnetic waves. Coaxial cables are used by the cable TV network and telephone companies.

Characteristics of Coaxial Cable

Different characteristics of coaxial cable are as follows:

- It is more expensive transmission medium than twisted pair.
- It provides higher data transfer rate than twisted pair cable.
- It can be installed very easily.

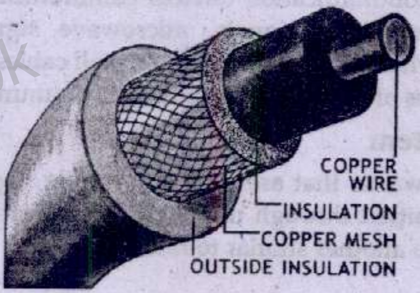


Figure: Coaxial Cable

13.23.1.3 Fiber Optics

Fiber optic cable consists of thin strands of glass or plastic called **core**. The strands are thin like human hair. The core is the center of the fiber optic cable that uses light to transmit data. Each strand is surrounded by a layer of glass inside the fiber optic cable called **cladding**. The cladding is further protected by a plastic coating called **jacket**. Most telephone companies, ISPs and cable TV operators are using fiber optics in their networks.

Advantages of Fiber Optics

- Fiber optic networks work at a very high speed.
- The information carrying capacity of fiber optics is very high.
- Fiber optic is lighter and smaller in size.
- It is more secure and reliable form of data transmission.
- It is not affected by electromagnetic waves.

Disadvantages of Fiber Optics

- It is difficult to install.
- It is expensive.

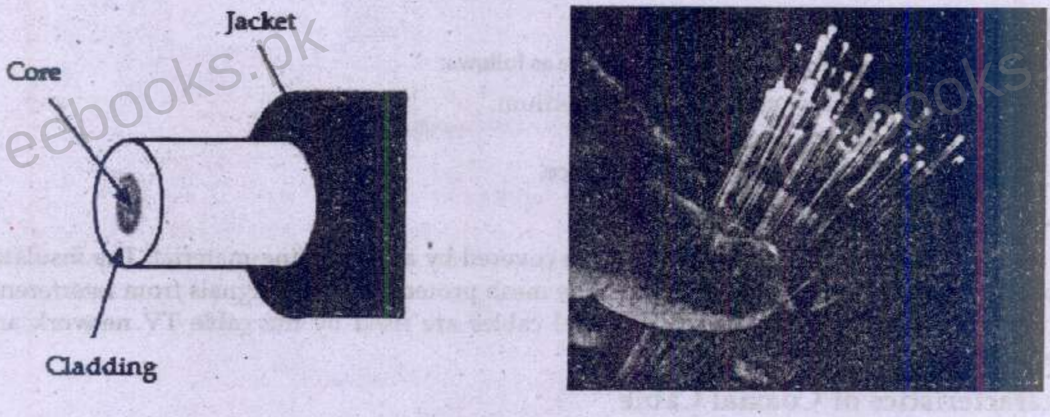

Jacket
Core
Cladding

Figure: Fiber Optics

13.23.2 Wireless Transmission Media

In unbounded media, communication devices communicate with each other through air or space using broadcast radio signals, microwave signals and infrared signals. Unbounded media are used where it is impossible to install cables. Data can be transferred all over the world. Some examples of unbounded media for communication are as follows:

13.23.2.1 Microwave System

Microwaves are radio waves that are used to provide high-speed transmission. Both voice and data can be transmitted through microwave. Data is transmitted through the air from one microwave station to another similar to radio signals.

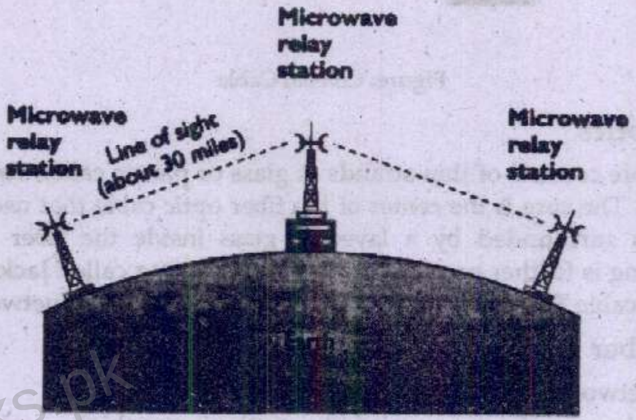

Microwave relay station
Microwave relay station
Line of sight (about 30 miles)
Microwave relay station
Earth

Figure: Microwave System

Microwave uses line-of-sight transmission. It means that signals travel in straight path and cannot bend. The microwave stations or antennas are usually installed on high towers or buildings. The stations are placed within 20 to 30 miles to each other. Each station receives signal from previous station and transfer to the next station. In this way, data transferred from one place to another. There should be no buildings or mountains between these stations.

13.23.2.2 Satellite Communication

Satellite communication is a space station that receives microwave signals from earth-based station. It amplifies the signals and retransmits them back to different earth-based stations. The satellite communication is placed about 22,300 miles above the earth. The transmission from earth-based station to satellite is called **uplink**. The transmission from satellite to earth-based station is called **downlink**.

The data transmission of satellite communication is very high. However, bad weather can affect the quality of satellite transmission. Satellite communication is used in different applications such as television and radio broadcasts, weather forecasting, global positing system and Internet connections etc.

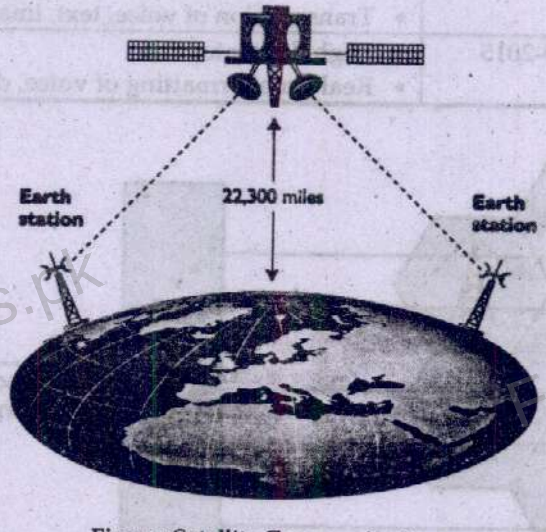


Figure: Satellite Communications

13.23.2.3 Broadcast Radio

Broadcast radio is a wireless transmission medium. It distributes radio signals through the air over long and short distances. Radio transmission requires a transmitter to send broadcast radio signals and a receiver to receive it. The receiver uses an antenna to receive the signals. Some networks use **transceiver** that can both send and receive signals.

An example of short-range broadcast radio communications is **Bluetooth™**. It uses short-range radio waves to transmit data at a rate of 1 Mbps among Bluetooth™ - enabled devices. Bluetooth™ is used in personal computers, Internet appliances, cellular phones, fax machines and printers.

13.23.2.4 Cellular Communication

Cellular communication is a wireless communication system. It is widely used all over the world for mobile communication. A cellular communication system is used to transmit text, images, voice and video.

Cellular communication divides a physical region into different sections known as cells. For example, a city can be divided into small cells. Each cell has a low-powered radio transmitter/receiver. The cells are close enough to one another so that the signal strength is maintained throughout the area. These cells are linked together to enable a large number of cellular phones to communicate with one another.

Different generations of cellular transmission are as follows:

Technology	Year	Features
1G	1981	• Make calls without operator assistance
2G	1991	• Decreased interference in digital signals • Improved reception • Improved security against cell phone fraud
3G	2001	• Faster data transmission • Better network capacity • Advanced network services • Transmission of voice, text, images and video
4G	2010-2015	• High data rates • Real-time formatting of voice, data, multimedia

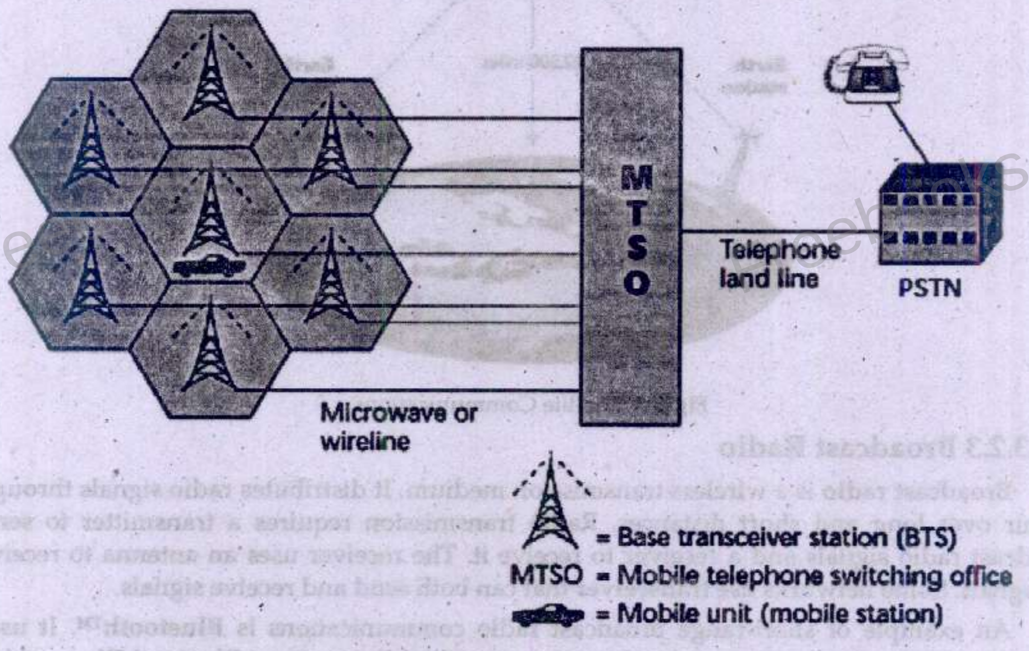

MTSO
Telephone land line
PSTN
Microwave or wireline
= Base transceiver station (BTS)
MTSO = Mobile telephone switching office
= Mobile unit (mobile station)

Figure: Cellular Communications

Components of Cellular Communication System

Different components of cellular communication system are as follows:

1. **MU** MU stands for **Mobile Unit**. The mobile unit consists of a control unit and a transceiver. The transceiver transmits and receives radio transmissions to and from a cell site.
2. **RBS** RBS stands for **Radio Base Station**. Radio base station is the physical location that provides coverage within a cell.
3. **PSTN** PSTN stands for **Public Switched Telephone Network**. It is the worldwide telephone network used for data and voice communication. It interconnects telephones and other communication devices on a worldwide basis.

4. MTSO MTSO stands for **Mobile Telephone Switching Office.** It controls the entire operation of a cellular system. It monitors the relative signal strength of cellular phone as reported by each of the cell tower. It switches the conversation to the cell tower that will give the best possible reception. It also has the mobile switching center that controls calls, provides billing information and locates cellular subscribers etc.

13.23.2.5 Infrared

Infrared (IR) is a wireless transmission medium. It sends signals using infrared light waves. It requires a line-of-sight transmission. Mouse, printer and digital camera have an IrDA port that enables the transfer of data from one device to another using infrared light waves. It is an alternative to short-range radio communications like Bluetooth™.

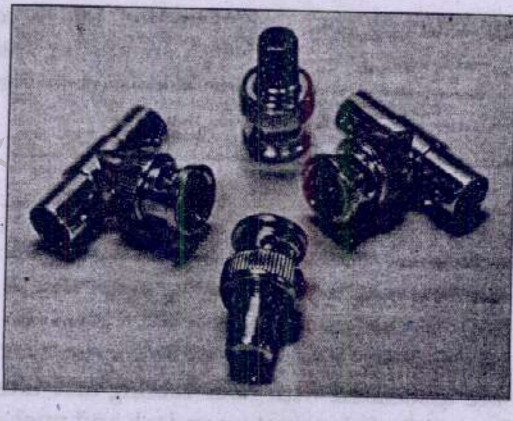
Transmission Media	Maximum Transfer rate
Infrared	115 kbps to 4 Mbps
Broadcast Radio	1 to Mbps
Bluetooth	
Cellular Radio	
2 G	9.6 kbps to 19.2 kbps
3 G	144 kbps to 2.4 Mbps
Microwave Radio	150 Mbps
Communication Satellite	1 Gbps

13.24 Media Connectors

Different types of media connectors are used with network media. Media connectors attach to the transmission media and allow physical connection to the computing device. Different media connectors are as follows:

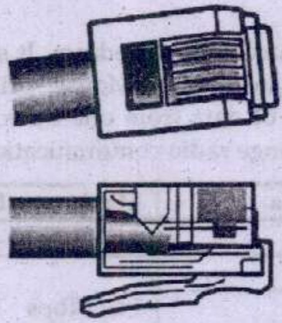
1. BNC Connectors

BNC connectors are associated with coaxial media and 10Base2 networks. BNC connectors are not very common but are still used on older network cards, and older hubs. The common BNC connectors are barrel connector, T-connector and terminators. The following figure shows two terminators and two T-connectors.



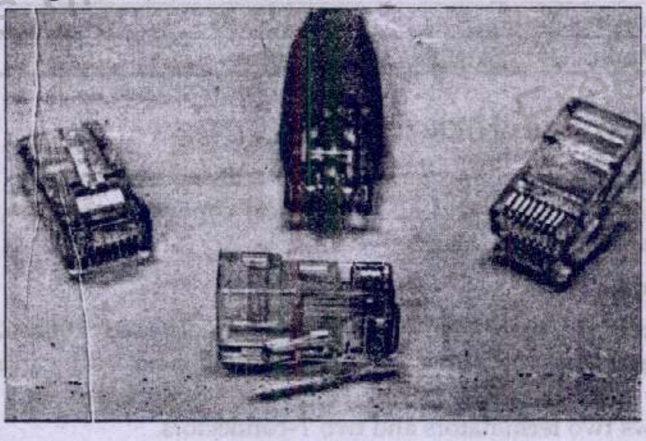
2. RJ-11 Connectors

RJ stands for **Registered Jack. RJ-11** connectors are small plastic connectors used on telephone cables. They have capacity for six small pins but all pins are not used in many cases. For example, a standard telephone connection only uses two pins and a cable used for a DSL modem connection uses four. RJ-11 connectors have small plastic flange on top of the connector to ensure a secure connection.



3. RJ-45 Connectors

RJ-45 connectors are used with twisted-pair cabling. They support up to eight wires. RJ-45 connectors are larger than RJ-11 connectors.

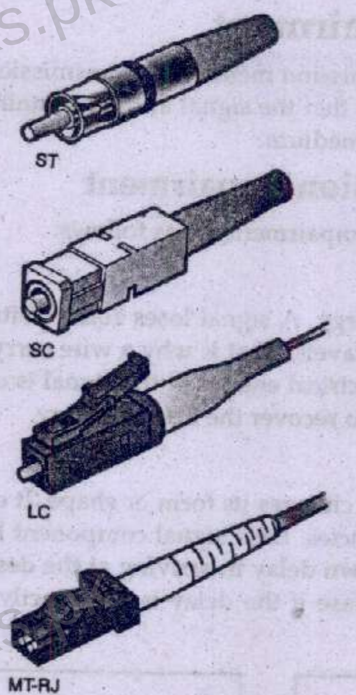


5. F-Type

F-Type connectors are screw on connections used to attach coaxial cable to devices. These connectors are most commonly associated with connecting Internet modems to cable or satellite Internet provider's equipment. They are also used to connect some proprietary peripherals.

6. Fiber Connectors

Different connectors are associated with fiber cabling and there are different ways to connect these connectors. These include bayonet, snap-lock and push-pull connectors.



The ST connector uses a half-twist bayonet type of lock.

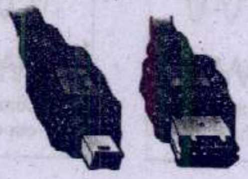
The SC uses a push-pull connector similiar to common audio and video plugs and sockets.

LC connectors have a flange on top, similar to an RJ-45 connector, that aids secure connection.

MT-RJ is a popular connector for two fibers in a very small form factor.

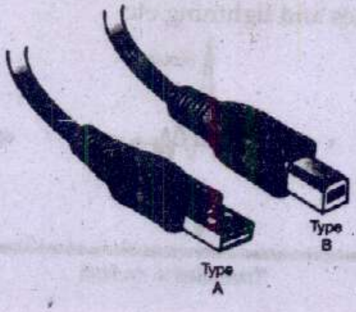
7. IEEE1394

IEEE1394 interface is also known as FireWire. It is commonly used to attach peripheral devices such as digital cameras or printers. However, it is possible to create small networks with IEEE1394 cables. The IEEE1394 interface comes in a 4 or 6-pin version.



8. Universal Serial Bus (USB)

Universal Serial Bus (USB) ports are very commonly used on both desktop and laptop computers. USB is used to connect peripherals such as MP3 players and digital cameras. Many manufacturers now make wireless network cards that plug directly into a USB port. Most desktop and laptop computers have 2 to 4 USB ports. However, USB hubs can be used for additional ports. Two types of connectors associated with USB ports are Type A and Type B. Type A connectors are more commonly used.



Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

13.25 Transmission Impairment

The signals travel through transmission media. The transmission media are not perfect and cause **signal impairment**. It means that the signal at the beginning of the medium is not the same as the signal at the end of the medium.

13.25.1 Causes of Transmission Impairment

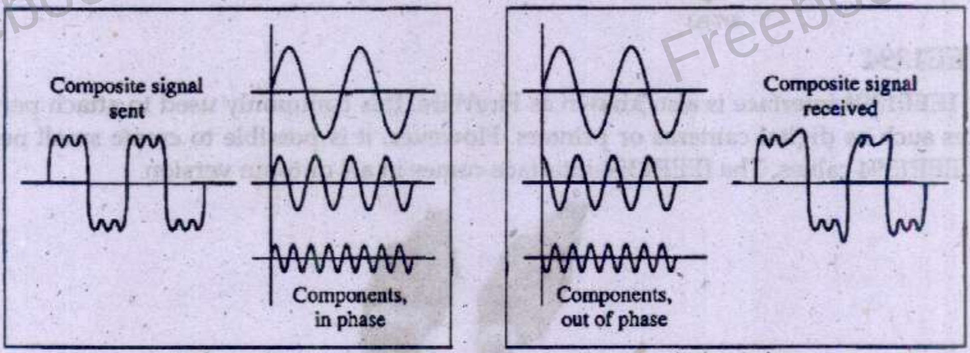
Different causes of transmission impairment are as follows:

1. Attenuation

Attenuation means a loss of energy. A signal loses some of its energy in overcoming the resistance of the medium when it travels. That is why a wire carrying electric signals gets warm after some time. Some of the electrical energy in the signal is converted into heat. The amplifier is used to amplify the signal to recover the loss of energy.

2. Distortion

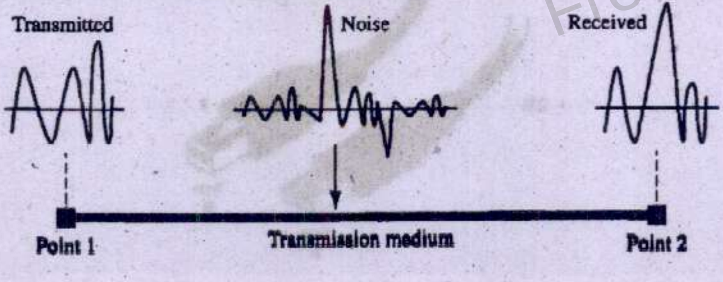
Distortion means that the signal changes its form or shape. It can occur in a composite signal that consists of different frequencies. Each signal component has its own propagation speed through a medium and has its own delay in arriving at the destination. This difference in delay may create a difference in phase if the delay is not exactly the same as the period duration.



At the sender At the receiver

3. Noise

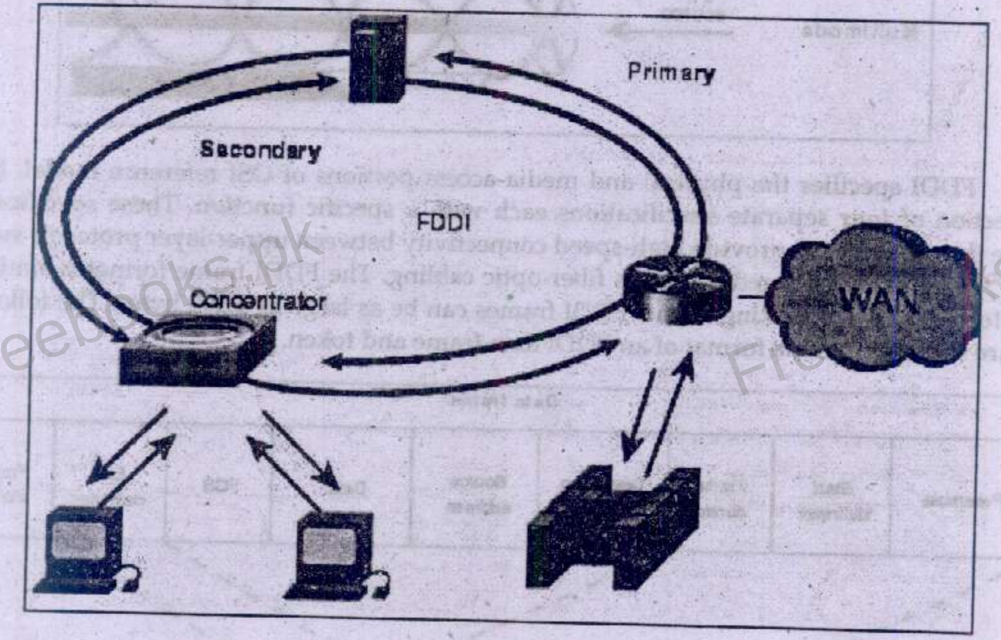
Any unwanted signal is called **noise.** Noise can be of different types like thermal noise, induced noise, crosstalk and impulse noise. The noise may corrupt the signal. **Thermal noise** is the random motion of electrons in a wire that creates an extra signal not originally sent by the sender. **Induced noise** comes from sources such as motors and appliances. **Crosstalk** is the effect of one wire on the other. **Impulse noise** is a signal with high energy in a very short time that comes from power lines and lightning etc.



13.26 Fiber Distributed Data Interface (FDDI)

The **Fiber Distributed Data Interface (FDDI)** specifies 100-Mbps token-passing, dual-ring LAN using fiber-optic cable. It supports high bandwidth and greater distances than copper. It is frequently used as a high-speed backbone technology.

FDDI uses dual-ring architecture with traffic on each ring flowing in opposite directions called **counter-rotating**. The dual rings consist of a primary and a secondary ring. The primary ring is used for data transmission during normal operation. The secondary ring remains idle. However, the secondary takes over the primary's functionalities if the primary does not work. The primary purpose of the dual rings is to provide superior reliability.

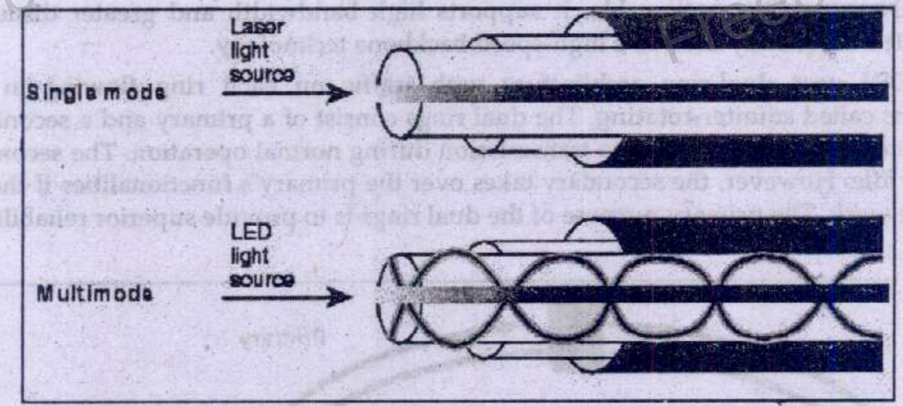


FDDI uses optical fiber as the primary transmission medium. It can also run over copper cabling. FDDI over copper is called **Copper-Distributed Data Interface (CDDI)**. Optical fiber has several advantages over copper media. It provides better security, reliability and performance because fiber does not emit electrical signals. A physical medium that emits electrical signals can be tapped. Therefore, it may permit unauthorized access to the data that is transiting the medium.

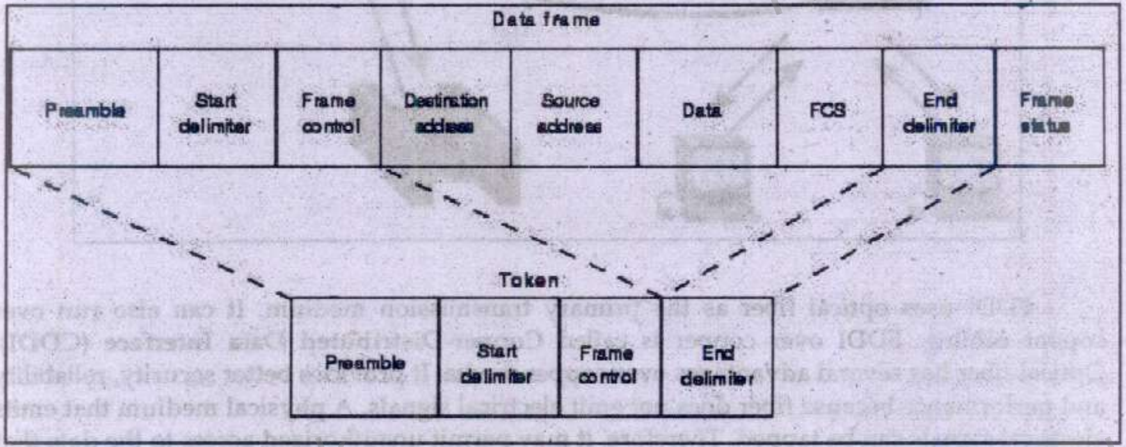
The fiber is also immune to electrical interference from radio frequency interference (RFI) and electromagnetic interference (EMI). Fiber supports much higher bandwidth than copper. Although recent technological advances have made copper capable of transmitting at 100 Mbps.

FDDI defines two types of optical fiber called **single-mode** and **multimode**. A **mode** is a ray of light that enters the fiber at a particular angle. Multimode fiber uses LED as the light-generating device while single-mode fiber generally uses lasers. **FDDI allows 2km between stations using multimode fiber and even longer distances using a single mode.**

The following figure shows single-mode fiber using a laser light source and multimode fiber using a light emitting diode (LED) light source.


Laser light source
Single mode
LED light source
Multimode

FDDI specifies the physical and media-access portions of OSI reference model. It is a collection of four separate specifications each with a specific function. These specifications have the capability to provide high-speed connectivity between upper-layer protocols such as TCP/IP and IPX and media such as fiber-optic cabling. The FDDI frame format is similar to the format of a Token Ring frame. FDDI frames can be as large as 4,500 bytes. The following figure shows the frame format of an FDDI data frame and token.


Data frame
Preamble
Start delimiter
Frame control
Destination address
Source address
Data
FCS
End delimiter
Frame status
Token
Preamble
Start delimiter
Frame control
End delimiter

Preamble	It gives a unique sequence that prepares each station for an upcoming frame.
Start delimiter	It indicates the beginning of a frame.
Frame control	It indicates the size of the address.
Destination address	It is the origin and destination address
Source address	It identifies the single station that sent the frame.
Frame check sequence	It is CRC error-checking
End delimiter	It is the end of the frame.
Frame status	It identifies whether the frame was recognized and copied by a receiving station.

13.27 OSI Model

OSI stands for **Open System Interconnection**. An open system is a set of protocols that allows any two systems to communicate even if their architectures are different. It covers all aspects of network communications. It was developed by the **International Standards Organization (ISO)** in 1983.

OSI model is layered framework for the design of network systems for communication between all types of computer systems. It consists of seven layers and each layer defines a part of the process of moving information across a network.

13.27.1 Advantages of OSI Model

Some advantages of OSI model are as follows:

- It divides the network communication process into smaller and simpler components.
- It allows multiple-vendor development via standardization of network components.
- It encourages industry standardization to define what functions occurs at each layer.
- It allows various types of network hardware and software to communicate.
- It prevents changes in one layer from affecting the other layers.

13.27.2 Layered Architecture

The OSI model consists of seven ordered layers. A message may pass through many intermediate nodes when it travels from one device to another. The intermediate nodes usually involve only the first three layers of the OSI model.

Each layer defines a set of functions which are different from other layers. With a single machine, each layer uses the services of the layer below it. For example, Layer 3 uses the services provided by layer 2 and it provides the services for layer 4. When two machines communicate, layer x on one machine communicates with layer x on the other machine. This communication takes place according to the protocols. The processes on each machine that communicate at a given layer are called **peer-to-peer processes**.

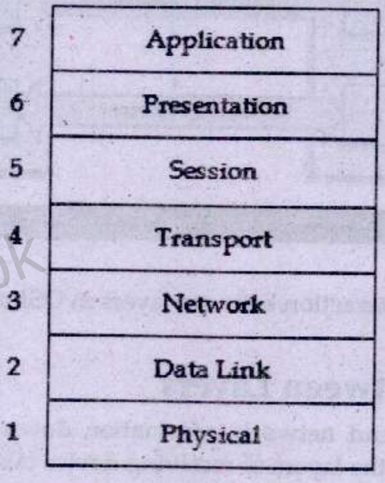
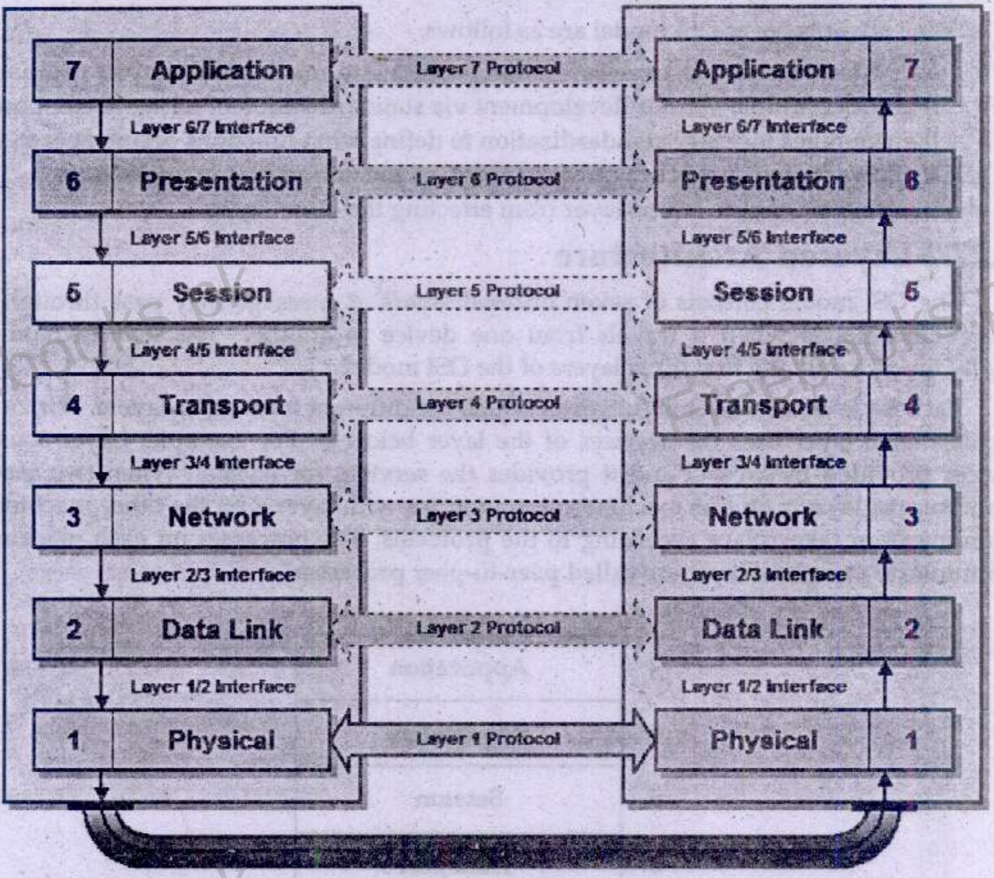


Figure: Seven Layers of OSI Model

13.27.3 Peer-to-Peer Processes

The communication at the physical layer is direct. However, the communication at the higher layers must move down through the layers on one device over to other device and then back up through the layers. Each layer in the sending device receives the message from the layer above it, adds its own information to the message and passes it to the layer below it.

At layer 1, the information is converted into a form that can be transmitted to the receiving device. The message is again converted at the receiving machine layer-by-layer. Each process receives and removes the data that is meant for it.


7 Application
Layer 7 Protocol
Application 7
Layer 6/7 Interface
Layer 6/7 Interface
6 Presentation
Layer 6 Protocol
Presentation 6
Layer 5/6 Interface
Layer 5/6 Interface
5 Session
Layer 5 Protocol
Session 5
Layer 4/5 Interface
Layer 4/5 Interface
4 Transport
Layer 4 Protocol
Transport 4
Layer 3/4 Interface
Layer 3/4 Interface
3 Network
Layer 3 Protocol
Network 3
Layer 2/3 Interface
Layer 2/3 Interface
2 Data Link
Layer 2 Protocol
Data Link 2
Layer 1/2 Interface
Layer 1/2 Interface
1 Physical
Layer 1 Protocol
Physical 1

Interaction between layers in OSI model

13.27.4 Interfaces between Layers

The passing of data and network information down through the layers of sending device and back up through the layers of receiving device takes place by an interface between each pair of adjacent layers. Each interface defines the information and services that must be provided by a layer for the layer above it. The well-defined interfaces and layer functions provide modularity to the network. The specific implementation of layer functions can be modified or replaced without any change to the surrounding layers.

13.27.5 Organization of the Layers

The seven layers can be divided into three groups:

- The first three layers are the network support layers. They deal with physical aspects of moving data from one device to other like electrical specifications, physical connections, physical addressing and transport timing and reliability.
- The last three layers are the user support layers. They provide interoperability among unrelated software systems.
- The layer 4 links both groups and ensures that the data transmitted from lower layers are in a form that can be used by upper layers.

The upper layers are implemented in software and lower layers are a combination of hardware and software except the physical layer. The physical layer is mostly the hardware.

The process of transmitting data starts from layer 7 which is the application layer. It moves from layer to layer in descending order. A header and possibly a trailer can be added to the data unit at each layer. The trailer is commonly added only at layer 2. The data is changed into an electromagnetic signal when it passes through the physical layer and then transported along a physical link.

The signal passes into layer 1 upon reaching its destination and is transformed back into digital form. The data moves up through the OSI layers. The header and trailer attached to the data at corresponding layer is removed from it as the data research the next higher layer. When the message reaches layer 7, it is in a form that is appropriate to the application and becomes available to the recipient.

13.27.6 Layers of OSI Model

The function of each layer (from bottom to top) is as follows:

1. Physical layer

Physical layer is the bottom layer of OSI model. It transmits stream of bits and defines how the data is transmitted over the network and what control signals are used. Its main function is to control how a stream of bits is sent and received over the physical medium. The common protocols used at this level are IEEE 802, IEEE 802.2, FDDI..

Physical layer must decide the following:

Characteristics of Media	The physical layer defines the characteristics and type of transmission medium.
Representation of Bits	The bits are encoded into electrical signals for transmission. The physical layer defines the type of encoding.
Data Rate	Physical layer defines the number of bits that will be sent in each second.
Synchronization	The clocks of sender and receiver are also synchronized.
Line Configuration	The physical layer defines the attachment of communication devices with medium.
Transmission Mode	The physical layer defines the direction of transmission between two devices: simplex, half-duplex or full-duplex.

2. Data Link Layer

The **Data Link Layer** is responsible for the reliability of the physical link established at layer 1.

Data link layer must decide the following:

Framing	The data link layer divides the stream of bits into manageable data units called frames.
Flow Control	If the rate of sending data is more than the rate of receiving data, data link layer imposes a flow control mechanism to prevent this situation.
Error Control	The data link layer detects and retransmits damaged or lost frames. It also prevents the duplication of frames.
Access Control	If two or more devices are connected to the same link, data link layer determines which device has control over the link at a given time.

Data Link Layer is divided in two sub-layers:

i. LLC

LLC stands for **Logical Link Control**. It is the upper sub-layer. It ensures reliability of physical connection. The standard protocol IEEE 802.2 is the most commonly used standard. **Point-to-Point Protocol** (PPP) is an important standard at this OSI level. It is used to communicate across point-to-point links. It is an important protocol for wide area networking.

ii. MAC

MAC stands for **Media Access Control**. It specifies how workstations cooperatively share the transmission medium. The IEEE 802.3 standard specifies a medium-access method known as "carrier sense multiple access with collision detection (CSMA/CD)."

3. Network Layer

The **network** layer is responsible for establishing, maintaining, and terminating network connections. It manages the delivery of data from source to destination.

Network layer determines logical path between the sender and the receiver. There may be many networks between two computers. This layer manages to send data from the source computer to the destination computer. The common protocols that operate at this level are IP, IPX, and X.25.

Some responsibilities of network layer are as follows:

Logical Addressing	The logical addressing implemented by data link layer handles the addressing problems locally. If a packet passes the network boundary, another addressing system is required to distinguish the source and destination systems. The network layer adds a header to the packet that comes from the upper layer. The header includes the logical addresses of the sender and receiver.
Routing	Routers or gateways are used to route packets to their final destination when independent networks are connected together to form an internetwork. This mechanism is provided by the network layer.

4. Transport Layer

The **transport layer** controls the flow of data. It ensures that messages are delivered error free. It divides large messages into small packets for efficient transmission. These packets are reassembled, checked for errors and acknowledged at receiving side. Data is retransmitted if there are errors in transmission. The common protocols that operate at transport layer are TCP, UDP, SPX and NetBEUI.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

Some responsibilities of transport layer are:

Service-point Addressing	Computer can run multiple processes at the same time. The source-to-destination delivery means the delivery from one computer to another and from one process of a computer to another process of another computer. Transport layer header includes service-point address or port address. The network layer gets each packet to correct computer. Transport layer gets entire message to the correct process on that computer.
Segmentation & Reassembly	A message is divided into segments. Each segment has a sequence number. Transport layer uses these numbers to reassemble the message correctly on destination computer.
Connection Control	Transport layer can be connection-oriented or connectionless. A connection-oriented transport layer establishes connection with transport layer of destination computer before delivering packets. The connection is terminated after delivering all data. A connectionless transport layer delivers each packet to transport layer of destination computer as independent packet.
Flow Control	Transport layer performs flow control end to end rather than across a single link.
Error Control	Transport layer performs error control end to end rather than across a single link. The sending transport layer ensures that the message is delivered to the receiving transport layer. Error correction is normally performed via retransmission.

5. Session Layer

The **session layer** establishes, manages, and terminates user connections. A session is an exchange of messages between computers. It synchronizes user tasks. Synchronization involves the use of **checkpoints** in the data stream. If a failure occurs, only the data from the last checkpoint is retransmitted.

Suppose we want to send 1000 pages of data. Checkpoint can be used after each 100 pages. If there is an error at page 320, the pages from 301 will be retransmitted. Page from 1 to 300 will not be retransmitted. Winsock and NetBIOS are usually shown as functioning at the session layer.

6. Presentation Layer

The presentation layer performs data reformatting, data compression and encryption. The common protocols that operate at the presentation layer include SMB, NCP, & NFS.

Data Reformatting	When two computers exchange data, the data is changed to bit streams before it is transmitted. Two computers may use different encoding techniques. The presentation layer at sending computer changes data according to the sender's format. The presentation layer at receiving computer changes the data according to the receiver's format.
Encryption	The presentation layer encrypts data before transmission. It means that the sender transforms original information to another form and sends the resulting message over the network. The receiver again transforms the message back to its original form. It is called decryption.

Compression	Data compression reduces the number of bits to be transmitted. The presentation layer compresses a large amount of data into small size.

7. Application Layer

The **application layer** is the top most layer of OSI model. It provides services directly to user applications. It enables the user to access the network. It provides user interfaces and support for services such as email, remote file access and transfer, shared database management, and other types of distributed information services.

File Transfer	It allows user to access, retrieve and manage files in remote computer.
Mail Services	It provides the basis for email forwarding and storage facilities.
Directory Services	It provides distributed database sources and access for global information about various objects and services.

13.28 TCP / IP Protocol Suite

Transmission Control Protocol / Internet Protocol (TCP/IP) is the protocol suite used for communications between hosts in most local networks and on the Internet. It can be used to enable network communications in local area networks and wide area networks as long as the hosts support the protocol. TCP/IP is widely supported and is included in operating systems such as Windows 7, XP, Vista, Mac OS, Linux, and Unix.

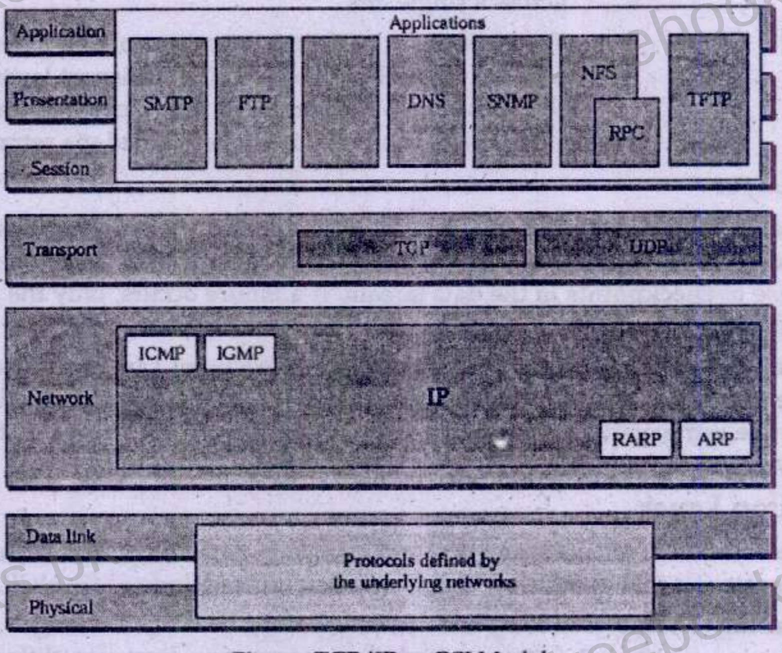


Figure: TCP/IP vs OSI Model

The TCP/IP protocol suite was devloped before the OSI model. Its layers are not same as OSI model. The original TCP/IP protocol suite was defined with four layers known as host-to-network, internet, transport and application. The host-to-network layer is equivalent to the combination of physical and data link layers of OSI model. The internet layer is equivalent to the network layer. The application layer performs the tasks similar to the tasks of session, presentation and application layers of OSI model. The transport layer in TCP/IP takes care of part of the duties of the session layer. So it is assumed that TCP/IP protocol suite consists of five layers called physical, data link, network, transport and application. The first four layers provide the physical standards, network interfaces, internetworking and transport functions which are equivalent to the first four layers of OSI model. However, the other three layers of OSI model are represented in TCP/IP by only the application layer.

TCP/IP is a hierarchical protocol. It consists of interactive modules. Each module provides a specific functionality but are not necessarily interdependent. OSI model specifies which functions belongs to each layer. The layers of TCP/IP protocol suite contain relatively independent protocols. The protocols can be mixed and matched according to system needs.

13.28.1 Physical and Data Link Layers

TCP/IP does not define any specific protocol at the physical and data link layers. It supports all the standard and proprietary protocols. A network in TCP/IP internetwork can be a local area network or wide area network.

13.28.2 Network Layer

TCP/IP network layer defines the protocols used for addressing and routing the data packets. The protocols that are part of TCP/IP network layer are IP, ARP, ICMP and IGMP.

Internetworking Protocol (IP)

The **Internetworking Protocol (IP)** is the transmission mechanism used by TCP/IP protocols. It is an unreliable and connection protocol also called **best-effort delivery** service. It means that IP provides no error checking or tracking. It does its best to transmit data to the destination without any guarantee.

IP transports data in packets called **datagrams**. Each packet is transported separately and can travel along different routes. It can arrive without sequence or can be duplicated. IP does not keep track of the routes and hos no facility to record datagrams once they arrive at the destination.

Address Resolution Protocol (ARP)

The **address resolution protocol** is used to associate a logical address with a physical address. Each device on a link is identified by a physical or station address on a typical physical network. It is usually stored on the network interface card (NIC). ARP is used to find the physical address of the node when its Internet address is known.

Reverse Address Resolution Protocol (RARP)

The **reverse address resolution protocol** allows a host to discover its Internet address when it only knows its physical address. It is used when a computer is connected to a network for the first time or when a diskless computer is booted.

ICMP Protocol

ICMP stands the **Internet Control Message Protocol**. It is used to control the flow of data in the network, reporting errors and performing diagnostics. A networking device like router sends an ICMP source-quench packet to a host that requests a slowdown in the data transfer. An important troubleshooting tool within the ICMP protocol is PING. It stands for **Packet InterNet Groper**. The ping command is used to verify connectivity with another host in the network. The destination host can be in a LAN or on the Internet. The ping command uses a series of echo requests. The networking device receiving the echo requests responds with a series of echo replies to test a network connection.

IGMP Protocol

IGMP stands for Internet Group Message Protocol. It is used when one host needs to send data to many destination hosts. It is called **multicasting**. The addresses used to send a multicast data packet are called **multicast addresses**. These are reserved addresses and are not assigned to hosts in a network. A router uses IGMP packets when it uses multicasting to share routing tables.

Another application to use IGMP packets is when a hosts wants to stream data to multiple hosts. Streaming means the data are sent without waiting for any acknowledgement that the data packets were delivered. The source in IGMP protocol does not care if the destination receives the packet. The streaming is an important application in the transfer of audio/video files over the Internet. Another feature of IGMP is that data is handed to the application layer as it arrives. This enables to begin processing the data for playback.

13.28.3 Transport Layer

The transport layer was represented in TCP/IP by two protocols called TCP and UDP. IP is a host-to-host protocol and can deliver a packet from one physical device to another. UDP and TCP are transport level protocols and are responsible for delivery of a message from one process to another process. A new transport layer protocol SCTP has been devised to meet the needs of some newer applications.

UDP

UDP stands for User Datagram Protocol and is a connectionless protocol. The UDP packets are transported over the network without establishing a connection and without any acknowledgement that the data packets arrived at the destination. It is useful in applications such as videoconferencing and audio feeds where acknowledgements are not necessary.

TCP

TCP stands for Transport Control Protocol and is a connection-oriented protocol. A connection oriented protocol establishes a network connection, manages the data transfer and terminates the connection. TCP protocol establishes a set of rules to establish the connection. It verifies the delivery of the data packets through the network and includes support for error checking and recovering lost data. It also specifies a procedure to terminating the network connection.

Stream Control Transmission Protocol (SCTP)

The stream control transmission protocol provides support for new applications such as voice over the Internet. It is a transport layer protocol that combines the best features of UDP and TCP.

13.28.4 Application Layer

The application layer in TCP/IP is equivalent to the combined session, presentation and application layers of OSI model. Many protocols are defined at this layer.

13.29 TCP/UDP Port Functions

Each TCP/IP protocol or application has a port associated with it. The target port number is checked to determine its related protocol or service when a communication is received. The request is then forwarded to that protocol or service. For example, HTTP is assigned port number is 80. A request from a Web browser for a web page is sent to port 80 on the target system. The target system receives the request and examines the port number. It sees port 80 and forwards the request to the Web server application. TCP/IP has 65,535 ports and the ports from 0 to 1023 are the well-known ports.

Some most common port assignments are as follows:

Protocol	Port Assignment
FTP	20
FTP	21
SSH	22
Telnet	23
SMTP	25
DNS	53
TFTP	69
HTTP	80
POP3	110
NNTP	119
NTP	123
IMAP4	143
HTTPS	443

13.30 IP Addressing

The MAC address provides the physical address for the network interface card. However, it does not provide any information about the location of its network. Internet Protocol (IP) addressing provides a solution to worldwide addressing by using a unique address that identifies the computer's local network. IP network numbers are assigned by **Internet Assigned Numbers Authority (IANA)**. This agency assigns IP addresses to computer networks. It ensures that no two different networks are assigned the same IP network address.

IP addresses are classified as IPv4 or IPv6. IP version 4 (IPv4) is the current TCP/IP addressing technique being used on the Internet. The address space for IPv4 is quickly running out due to the rapid growth of the Internet and the development of new Internet-compatible technologies. However, both IPv4 and IPv6 are supported by manufacturers of networking equipment and the latest computer operating systems.

13.30.1 IPv4 Addressing

IP version 4 (IPv4) is the current TCP/IP addressing technique being used on the Internet. It is also the preferred technique for providing addresses in the LAN. A typical IP address is written as **216.27.61.137**. The IP address is written in decimal format. But IP address in computer is written in binary form as follows:

11011000.00011011.00111101.10001001

The four numbers in an IP address are called **octets** because each part contains eight positions in binary form. If all positions are added together, it gives 32. That is why IP addresses are considered 32-bit numbers.

Each position in an octect can have two different states (1 or 0). It means that the total possible combinations per octet is 2^8 or 256. Each octet can contain any value between 0 and 255. A combination of four octets gives 2^{32} or a possible 4,294,967,296 unique values.

Some certain values are restricted from use as typical IP addresses. For example, the IP address 0.0.0.0 is reserved for the **default network** and the address 255.255.255.255 is used for **broadcasts**.

The octets are used to create classes of IP addresses that can be assigned to a particular business, government or other entity based on size and need. The octets are divided into two sections of **Net** and **Host**. The Net section always contains the first octet. It is used to identify the network in which a computer exists. Host (or **Node**) identifies the actual computer on the network. The Host section always contains the last octet.

There are five IP classes and certain special addresses:

<table>
<tr><th>Class</th><th>1st Octet</th><th>2nd Octet</th><th>3rd Octet</th><th>4th Octet</th></tr>
<tr><td>A</td><td>0 Net ID</td><td colspan="3">Host ID</td></tr>
<tr><td>B</td><td colspan="2">10 Net ID</td><td colspan="2">Host ID</td></tr>
<tr><td>C</td><td colspan="3">110 Net ID</td><td>Host ID</td></tr>
<tr><td>D</td><td colspan="4">1110 Multicast address</td></tr>
<tr><td>E</td><td colspan="4">1111 Reserved for future use</td></tr>
</table>

Class A

The **class A** is used for very large networks such as an international organization. The IP addresses of this class are from 1 to 126. The other three octets are used to identify each host. It means that there are 126 networks in Class A.

Each network has 16,777,214 (2^{24} - 2) possible hosts for a total of 2,147,483,648 (2^{31}) unique IP addresses. In Class A networks, the value of high order bit in the first octet is always 0.

Net	Host or Node
115.	24.53.107

- **Loopback Address:** The IP address 127.0.0.1 is used as loopback address. It means that it is used by the host computer to send a message back to itself. It is commonly used for troubleshooting and network testing.

Example

0nnnnnnn hhhhhhhh hhhhhhhh hhhhhhhh

- The first bit is 0
- The next 7 bits represent the network
- The next 24 bits represent the host
- The initial byte can be from 0 - 127
- Total address in Class A are 126 (0 and 127 are reserved)
- 16,777,214 hosts on each Class A

Class B

The **class B** is used for medium-sized networks. The IP addresses of this class are from 128 to 191. Class B addresses also include the second octet as part of Net identifier. The other two octets are used to identify each host. It means that there are 16,384 (2^{14}) networks in Class B. Each network has 65,534 (2^{16} - 2) possible hosts for a total of 1,073,741,824 (2^{30}) unique IP addresses. Class B networks have a first bit value of 1 and a second bit value of 0 in the first octet.

Net	Host or Node
145.24.	53.107

Example

10nnnnnn nnnnnnnn hhhhhhhh hhhhhhhh

- The first two bits are 10
- The next 14 bits represent the network
- The next 16 bits represent the host
- The initial bytes can be from 128 - 191
- Total addresses in Class B are 16,384
- 65,532 hosts are available on each Class B

Class C

The **class C** addresses are commonly used for small to medium-size organization. The first of the IP addresses of this class is from 192 to 223. Class C addresses also include the second and third octets as part of the Net identifier. The last octet is used to identify each host. It means that there are 2,097,152 (2^{21}) networks in class C. Each network has 254 (2^8 -2) possible hosts for a total of 536,870,912 (2^{29}) unique IP addresses. Class C networks have a first bit value of 1, second bit value of 1 and a third bit value of 0 in the first octet.

Net	Host or Node
195.24.53.	107

Example

110nnnnn nnnnnnnn nnnnnnnn hhhhhhhh

- The first three bits are 110
- The next 21 bits represent the network
- The next 8 bits represent the host
- The initial bytes can be from 192 - 223
- Total addresses in Class C are 2,097,152
- 254 hosts on each Class C

Class D

The **class D** is used for multicasts. **Multicasting** is used to pass the copies of datagram to a selected group of hosts instead of individual host. Class D is slightly different from the first three classes. It has a first bit value of 1, second bit value of 1, third bit value of 1 and fourth bit value of 0. The other 28 bits are used to identify the group of computers where the multicast message will be passed.

Net	Host or Node
224.	24.53.107

Example

1110mmmm mmmmmmmm mmmmmmmm mmmmmmmm

- The first four bits are 1110
- The next 28 bits are for multicast address
- The initial bytes can be from 224 - 247

Class E

The **class E** is used for experimental purposes only. It is also different from first three classes. It has a first bit value of 1, second bit value of 1, third bit value of 1 and fourth bit value of 1. The other 28 bits are used to identify the group of computers where the multicast message will be passed.

Net	Host or Node
240.	24.53.107

- **Broadcast:** The messages that are intended for all computers on a network are sent as broadcasts. These messages always use the IP address **255.255.255.255.**

Example

1111rrrr rrrrrrrr rrrrrrrr rrrrrrrr

- The first four bits are 1111
- The next 28 bits are reserved address
- The initial byte can be from 248 – 255
- It is reserved for experimental use

13.30.2 IPv6 Addressing

IPv4 is the current TCP/IP addressing technique used on the Internet. The address space for IPv4 is quickly running out due to the rapid growth of the Internet. **IPv6** is the proposed solution for expanding the possible number of users on the Internet. It is also called **Ipng** or the next generation IP.

IPv6 uses 128-bit address technique as compared to IPv4's 32-bit address structure. It provides a large number of IP addresses (2^{128}). IPv6 numbers are written in hexadecimal rather than dotted decimal. An example of a 32-hexadecimal digit IPv6 address is as follows:

6789:ABCD:1234:EF98:7654:321F:EDCB:AF21

The important changes from IPv4 to IPv6 are as follows:

Expanded Address Space	The current Internet addresses are 32 bits long. It limits the number of unique addresses to approximately 4 billion. Some addresses are reserved such as broadcast addresses and some addresses in a class may not be used. IPv6 increases the size of address field to 128 bits. It allows up to 3 x 1038 addresses.
Quality of Service (QoS)	IPv6 will institute service categories to prioritize the flow of data to accommodate time-sensitive transmissions such as audio and video. The real-time transmissions will be provided with improved performance. The priority designations include the classifications such as follows:

- Uncharacterized traffic
- Filler traffic such as net news
- Unattended data transfer such as e-mail
- Bulk transfer such as file transfers
- Interactive transfers
- Real-time transfers

IP Header Changes	The format of IP header will change to provide larger address space and quality of service. The header fields that are currently not used will be dropped. The header extensions are allowed to provide variable-length headers. The extension headers can provide functions such as security, integrity, destination-specific information and routing control.
Security & Privacy Changes	IPv6 will allow extensions to the header for secure transmissions. The extensions will provide many authentication algorithms to detect and eliminate security problems.
Interoperability with IPv4	The new version will be backward compatible with the current version. The implementation of IPv6 may be accomplished incrementally to allow the nodes to be upgraded to the new version.

Types of IPv6 addresses

Different types of IPv6 addresses are as follows:

Unicast	The unicast IPv6 address is used to identify single network interface address. Data packets are sent directly to the computer with specified IPv6 address.
Multicast	IPv6 addresses are defined for a group of networking devices. Data packets sent to a multicast address are sent to the entire group of networking devices such as a group of routers running the same routing protocol. Multicast addresses start with the prefix FF00::/8. The next group of characters in the IPv6 multicast address are called the scope. The scope bits are used to identify which ISP should carry the data traffic.
Anycast	The anycast IPv6 address is obtained from a list of addresses but is only delivered to the nearest node.

13.30.3 IP Address Assignment

IP address allocation is governed by **Internet Assigned Number Authority (IANA)**. IANA delegates the allocation to the **regional Internet registries (RIRs)** to coordinate the global IP allocation more effectively. Each regional Internet registry is responsible for a different area.

The five RIRs for different regions are as follows:

- AfriNIC for Africa Region
- APNIC for Asica / Pacific Region
- ARIN for North America Region
- LACNIC for Latin America and some Caribbean Islands
- RIPE NCC for Europe, the Middle East and Central Asia

These RIRs assign block of IP addresses to the qualifying ISPs and end users. The ISPs then issue addresses to their customers.

13.31 Protocols

Different important protocols include TCP, UDP, DHCP, SMTP, SNMP, FTP, SFTP, TFTP, POP3, IMAP4, HTTP, HTTPS, NAT and ICS.

13.31.1 TCP

TCP stands for **transmission control protocol**. It provides full transport layer services to applications. It is a reliable stream transport port-to-port protocol. The term **stream** means connection-oriented. It means that the sender cannot send a message until the receiver is contacted and agrees to a communication. TCP generates a virtual circuit between sender and receiver by creating this connection. The connection remains active for the duration of a transmission. In this way , the receiver knows to expect the entire transmission rather than a single packet.

TCP is a connection oriented service. It is responsible for the reliable delivery of the entire stream of bits contained in the message generated by the sending application. Reliability is ensured by providing error detection and retransmission of damaged frames. All segments must be received and acknowledged before the transmission is considered complete and the virtual circuit is discarded.

At the sending end, TCP divides long transmission into smaller data units and packages each into a frame called a **segment**. Each segment includes a sequencing number for reordering after receipt and an acknowledgement id number. Segments are carried across network links inside of the IP datagram. At the receiving end, TCP collects each datagram and reorders the transmission based on the sequence number.

TCP Features

Some important features of TCP are as follows:

1. Connection Opening

On the sending host, a process such as web browser issues a request to send data to a destination host such as web server. TCP creates an initial segment designed to open the connection between sender and receiver. In this initial contact, the two systems exchange IP addresses and port numbers and setup the flow control and sequencing.

2. Flow Control

The sending and receiving hosts exchange the number of bytes to be accepted at on time. In this way, the sender sends an amount of data that can be handled by the receiver. The amount of data can be changed as the circumstances change on each machine.

3. Sequencing

Every segment is assigned a sequence number. The sequence number is stored in the first byte in every segment. This technique is used by the receiving host to reassemble any segments that arrive out of order.

4. Acknowledgement

When TCP transmits a segment, it holds the segments in a queue until the receiving TCP issues an acknowledgement that the segment has been received. The sender retransmits the segment if it does not receive the acknowledgement.

5. Error Detection

A value in the header is used by the receiver to test the integrity of incoming segment. The receiver sends an error message to the send if the segment is corrupted. The sender then immediately retransmits the segment.

6. Connection Closing

The connection is closed when no more data is to be sent. The sending TCP sends a segment that tells the receiver that the socket should be closed.

13.31.2 UDP

UDP stands for **user datagram protocol**. It is a connectionless service. It means that the sender does not have to first establish a link to the receiver before transmitting data. UDP is a simple transport protocol. It adds only port addresses, error control and length information to the data from upper layer. The packet produced by UDP is called a user datagram.

A brief description of its fields is as follows:

Source port address	It is the address of application program that creates a message.
Destination port address	It is the address of application program that receives the message.
Total length	It defines the total length of user datagram in bytes.
Checksum	It is a 16-bit field that is used in error detection.

UDP provides only basic functions required for end-to-end delivery of transmission. It does not provide any sequencing or reordering functions. It cannot specify the damaged packet when an error occurs. UDP does not contain an ID or sequencing number for a particular datagram.

13.31.3 DHCP

Dynamic Host Configuration Protocol (DHCP) is the most frequently used method of dynamic IP addressing. The servers that manage the dynamic assignment of IP addresses are called **DHCP servers**. The workstations that are dynamically assigned IP addresses by DHCP servers are called **DHCP clients**. DHCP software resides on clients and servers to manage the dynamic assignment of IP addresses.

The range of IP addresses to be assigned to clients is specified when DHCP servers are configured. DHCP servers maintain a list of currently assigned IP addresses from assignable IP address range. A client can be assigned any IP address from assignable list.

DHCP server can assign an IP address in three different ways:

Automatic Allocation	In automatic allocation, the server attempts to always assign the same address to a given host or ode. It is useful for the hosts that provide services to the users such as mail servers. Such type of host needs the same address each time because other hosts may have saved its Internet address.
Dynamic Allocation	In dynamic allocation, an IP address is automatically assigned to a client in a TCP/IP network. It is typically assigned by a DHCP server each time the client logs on to the network.
Manual Allocation	In manual allocation, a network administrator assigns address to a particular host or node.

DHCP process starts when the client sends a broadcast message to locate DHCP server. The broadcast message contains client's ID, request for IP address and desired options such as subnet mask, DNS server, domain name and static route. The broadcast message that is sent is called **DHCPDISCOVER** message. Each DHCP server that receives this message can send **DHCPOFFER** message to the client to offer an IP address that is not currently in use. Typically, the least recently used address in the server's assignable address list is selected. The client indicates its acceptance of IP address by broadcasting **DHCPREQUEST** message. The server acknowledges client's acceptance of IP address in **DHCPACK** message. The message also delivers additional configuration information to the client.

13.31.4 SMTP

SMTP stands for Simple Mail Transfer Protocol. It is one of the most popular network services is electronic mail. It supports electronic mail on the Internet. It is a system for sending messages to other computer users based on email addresses. SMTP provides the facilities for mail exchange between users on same or different computers. It also supports:

- Sending a single message to one or more recipients
- Sending messages that include text, voice, video or graphics
- Sending messages to users on networks outside the Internet.

13.31.5 SNMP

SNMP stands for Simple Network Management Protocol. It is a management protocol that enables network devices to communicate information about their state to a central system. It also enables the central system to pass configuration parameters to the devices.

In an SNMP configuration, a system acts as the central communication point for all SNMP-enabled devices on the network. This system is known as a manager. A software called **SNMP agent** is set up and configured with IP address of the manager on each device to be managed and monitored via SNMP. The SNMP manager can communicate and retrieve information from the devices running SNMP agent software depending on the configuration. The agent can also communicate the occurrence of certain events to the SNMP manager as they happen. These messages are known as **traps**.

An important part of SNMP is an SNMP management system. It is a computer that runs a special type of software called **Network Management System (NMS)**. Most of NMS applications use graphical maps of the network to locate a device and then it. The queries are built into the application and are triggered by a click. The user can also issue SNMP requests from a command-line utility. An SNMP agent can be any device that can run a small software that provides the facility of communication with an SNMP manager.

13.31.6 File Transfer Protocol (FTP)

The **File Transfer Protocol** is an application layer protocol. It provides a method for uploading and downloading files from a remote system running FTP server software. FTP uses the TCP transport protocol to guarantee the delivery of data packets.

FTP provides some basic security capabilities such as user authentication. FTP server can be configured to accept anonymous logons instead of creating user accounts for every user. In this case, the username is anonymous and the password is normally the user's email address. Most FTP servers that provide files to the general public operate in this way.

FTP is popular for distributing files over the Internet. It can also be used within an organization to exchange large files frequently with other people or organizations.

13.31.7 Secure File Transfer Protocol (SFTP)

A problem associated with FTP is that it transmits data between sender and receiver in an unencrypted format. It is solved in another protocol called **Secure File Transfer Protocol**. SFTP is based on **Secure Shell (SSH)** technology. SSH provides authentication between sender and receiver in addition to encryption capabilities. SFTP is implemented through client and server software that is available for all commonly used computing platforms.

13.31.8 Trivial File Transfer Protocol (TFTP)

Trivial file transfer **protocol** is a variation of FTP. FTP and TFTP are both application layer protocols. However, TFTP does not have the security capability or level of functionality like FTP. It uses only UDP as a transport protocol and makes it a connectionless protocol. It has a lower overhead than FTP. TFTP does not provide directory navigation. FTP provides the commands to navigate around and manage the file system. However, TFTP does not provide such capability.

13.31.9 POP3 and IMAP4

POP3 stands for **Post Office Protocol version 3** and **IMAP4** stands for **Internet Message Access Protocol version 4**. Both POP3 and IMAP4 are used to download email from a mail server. The email is transported around the network via SMTP and stored on a server. It is then downloaded from the server using POP3 and IMAP4. A problem with POP3 is that the password for accessing mailbox is transferred across the network in clear text. It can be accessed by someone easily. IMAP uses more sophisticated authentication system that makes it difficult to access the password.

13.31.10 Hypertext Transfer Protocol (HTTP)

Hypertext transfer protocol is used to download text, graphics, multimedia and other material a Web server. It defines which actions can be requested by clients and how servers should answer these requests. HTTP uses TCP as a transport protocol that makes it a connection-oriented protocol. It can also use UDP for certain functions.

HTTP uses a **uniform resource locator (URL)** to determine which page should be downloaded from the server. The URL contains the type of request, name of the server being contacted and the page being requested. HTTP works at the application layer of OSI model.

13.31.11 Hypertext Transfer Protocol Secure (HTTPS)

HTTP requests are normally sent in clear text. It is critical security problem for some Internet transactions such as online banking or e-commerce. **HTTPS** protocol is a solution for such applications. It uses a security technology known as **Secure Sockets Layer (SSL)**. This technology encrypts the information sent between the client and the host. HTTPS uses TCP transport protocol and operates at the application layer of the OSI model. The URL displays **https://** in the address bar instead of **http://** when using HTTPS protocol. An example of an HTTPS URL address is https://www.anyweb.com.

13.32 NAT

NAT stands for **Network Address Translation**. The basic principle of NAT is that many computers can hide behind a single registered IP address or a group of registered IP addresses. It means that only one registered IP address is required on the external interface of the system. However, the system can be working as a gateway between internal private network and external public network such as the Internet.

A system performing the NAT service transfers the requests to the external network. For example, a client requests a website and the request goes through the NAT server to the Internet. It appears to the remote system that the request is coming from a single address of the NAT server not from the individual client systems. A system that performs NAT function keeps track of all requests. It directs the required data to the system that requested for it.

The servers that provide NAT functionality work in different ways. For example, it is possible to statically map a single internal IP address to a single external IT address. In this case, the outgoing requests always appear from the same IP address. Alternatively, it can work with a group of public IP addresses. In this case, the NAT system assigns addresses to devices on a first-come first-serve basis.

13.33 ICS

ICS stands for **Internet Connection Sharing**. It is an implementation of NAT on Windows platforms since Windows Me. ICS makes it very simple to share an Internet connection with multiple systems on the network. The configuration of ICS is very simple because it was intended for small office network or a home network to share a single Internet connection. However, it provides no security and the system providing shared connection is not secure against outside attacks. That is why, ICS should be used only in conjunction with a firewall application.

13.34 DNS

Domain Name System consists of name resolution software to locate computers on the Internet and other TCP/IP networks by domain name. It maps host or node address to domain name. The mapping is provided by DNS servers. The DNS servers enable Internet users to reach another host or node without providing an IP address. The user can type a URL or click on a hyperlink to reach a particular Web page. It is not necessary to know the IP address of the Web server on which the Web page is stored.

DNS uses a database that contains domain names and associated IP addresses. Domain names are used as indexes in DNS database. The DNS database has an entry for each domain name. It identifies DNS server that has the most detailed information about the host or node such as its IP address and routing information. DNS servers communicate with one another to keep the database current. A DNS server sends query to other DNS servers for information if it does not know the address of a particular host or node specified in a URL. It forwards the information is to the Internet user's computer when it is found. The user's computer receives the information about the destination host or node's IP address. It can then build the TCP/IP packet required to send a message or request to the destination host/node.

13.35 Static and Dynamic IP Addresses

Static IP addresses are the IP addresses that are assigned to an organization's hosts and nodes permanently. The servers and routers are usually assigned static IP addresses. The users connected to the Internet via cable modems and DSL may also be assigned static IP addresses by ISPs.

Dynamic IP address is an IP address that is automatically assigned to a client in a TCP/IP network. It is typically assigned by a DHCP server each time the client logs on to the network. A client's IP address may vary across communication sessions when dynamic IP addressing is used.

13.36 Firewalls

Firewall is a set of components used to restrict the access to data and information on a network. It may consist of hardware and software. Firewalls are used to restrict the unauthorized users from accessing the resources of an organization.

Firewalls restrict data flows between the Internet and protected systems. It can be is done by defining the traffic such as users or addresses that is allowed and disallowing all other communication. Another way is to define what is prohibited and allowing all other transmissions. The primary purpose of a firewall is to intercept each packet addressed to the network and to decide whether to pass it on to the destination host. Packets originating from unknown or unauthorized IP addresses are carefully scrutinized. These packets are blocked if there are sufficient red flags. Firewall software can be implemented in routers, dedicated hosts or in other network technologies including LAN hubs/switches and network adapter cards. The major kinds of firewalls include packet filtering and proxy firewalls.

Packet Filtering

In **packet filtering**, a limit is placed on the packets that can enter the network. It can also limit information moving from one segment to another. ACLs are used to enable the firewall to accept or deny data packets. The disadvantages of packet filtering are as follows:

- Packets can still enter the network by fragmenting the data packets.
- It is difficult to implement complex ACLs.
- Not all network services can be filtered.

Proxy Firewalls

A **proxy server** is used by clients to communicate with secure systems using a proxy. The client gets access to the network via the proxy server. This step is used to authenticate the user, establish the session and set policies. The client must connect to the proxy server to connect to resources outside the network. The disadvantages of the proxy server are

- The proxy server can run very slowly.
- Adding services can be difficult.
- There can be a potential problem with network failure if the proxy server fails or is corrupted.

Review Questions

1. Define data communication.

Data communication is a process of transferring data electronically from one place to another. Data can be transferred by using different media. Data may consist of text, graphics and sound etc.

2. Define signal and signaling.

An electromagnetic or light wave used to transmit data from one place to another is called a signal. Propagation of signals across a communication medium is called signaling.

3. List out different elements of data communication.

The basic elements of data communication are as follows:

Sender: Sender is a device that sends the message. It is also called source or transmitter.

Medium: Medium is the physical path that connects the sender and the receiver. It is used to transmit data.

Receiver: Receiver is a device that receives message. It is also called sink.

Transmission Medium: Transmission medium is the path through which the messages are transferred.

Protocol: A protocol is a set of rules that governs data communication. It represents an agreement between communication devices.

4. Differentiate between sender and receiver.

Sender is a device that sends the data message. It is also called source. Sender is normally a computer. Receiver is a device that receives message. It can be computer, printer, or another computer related device. The receiver must be capable of accepting the message.

5. Define transmission modes.

The way in which data is transmitted from one place to another is called data transmission modes. Simplex, half duplex and full duplex are modes of data transmission.

6. Describe simplex mode of data transmission.

In simplex mode, data can flow only in one direction. It operates in a manner similar to a one-way street. The direction of flow never changes. A device with simplex mode can either send or receive data. Traditional television broadcast is an example of simples mode.

7. Describe half duplex mode of data transmission.

In half-duplex mode, data can flow in both directions but not at the same time. It is transmitted one-way at one time. A device with half-duplex mode can send or receive data but not at the same time. So, the speed of half-duplex mode is slow. Internet surfing is an example of half duplex transmission.

8. Compare simplex and half duplex mode.

In simplex mode, data can flow only in one direction. Radio and television broadcasting are the examples of this mode. In half-duplex mode, data can flow in both directions but not at the same time.

9. What is analog and digital data?

Data can be analog or digital. The term analog data refers to information that is continuous. Analog data take on continuous values. The human voice is an example of analog data. Digital data refers to information that has discrete states. Digital data take on discrete values. Data stored in the memory of a computer in the form of 0s and 1s is an example of digital data.

10. What is analog and digital signal?

The analog signal is a variation in a smooth way over time e.g. speech. Analog signals can have an infinite number of values in a range. Digital signal maintains a constant level then changes to another constant level. e.g., binary 1's and 0's. Digital signals can have only a limited number of values.

11. Write are two characteristics of analog signals.

Two characteristics of an analog wave are frequency and amplitude. The number of times a wave repeats during a specific time interval is known as frequency. The height of wave within a given period of time is known as amplitude.

12. Differentiate between analog and digital signal. Which is popular?

Analog signals can have an infinite number of values in a range whereas digital signals can have only a limited number of values. Digital signals are popular because they provide lower error rates, higher transmission speed and quality voice transmission.

13. What is a periodic and nonperiodic signal?

Both analog and digital signals can be periodic or nonperiodic signals. A periodic signal completes a pattern in a definite time frame is called periodic. We can say that periodic signal repeats itself after interval of time. The completion of a patteren is called cycle. A nonperiodic or aperiodic signal does not repeat itself after equal intervals of time.

14. What do you know about parallel data transmission?

A method of transmission in which groups of bits are sent at same time over multiple wires is called parallel transmission. It is usually unidirectional. Each bit is transmitted over a separate line. Data transmission between computer and printer is a parallel transmission.

15. What is serial data transmission?

A method of transmission in which data is sent one bit at a time is called serial transmission. Telephone lines use this method of data transmission. Each individual bit of information travels along its own communication path.

16. Differentiate between serial and parallel transmission.

Parallel transmission is faster because all bits are sent at the same time. Serial transmission is slower than parallel transmission because data is sent sequentially one bit at a time. Each individual bit of information travels along its own communication path.

17. Describe Asynchronous data transmission.

In **asynchronous transmission,** data is transmitted character by character. There are irregular gaps between characters. It uses flow control instead of clock to synchronize data between source and destination. It is cheaper because data is not saved before it is sent.

18. Describe Synchronous data transmission.

In synchronous mode, saved data is transmitted block by block. Each block may consist of many characters. It uses a clock to control the timing of bits being sent. Synchronous transmission is much faster than asynchronous because there is no gap between characters.

19. How does asynchronous transmission take place?

In asynchronous transmission, data is transmitted character by character. It uses a special start signal. A start bit has a value of 0. The value of 0 indicates that a character is about to be transferred. It alerts the receiver and it gets ready to receive the character. If start bit has a value 1, it indicates that the line is idle.

20. Compare asynchronous and synchronous transmission.

In asynchronous transmission, data is transmitted one character at a time. The sender and receiver are not synchronized with each other. Synchronous transmission sends a block of characters at a time. It allows sender and receiver to be synchronized with each other. Synchronous transmission is typically more efficient than asynchronous communications.

21. Define a computer network.

A computer network consists of two or more computers that are connected together to share information and resources. The computers in the network are connected together through communication media. The computers in the network can be in the same room, building or at different places.

22. What are the criteria necessary for an effective and efficient network?

Performance: It can be measured in many ways, including transmit time and response time.

Reliability: It is measured by frequency of failure, the time it takes a link to recover from a failure, and the network's robustness.

Security: Security issues includes protecting data from unauthorized access and viruses.

23. Name the factors that affect the performance of the network.

The factors that affect the performance of the network include number of users, type of transmission medium, hardware and software.

24. How the computers in network are connected?

The computers in the network are connected together through communication media. The communication media can be a physical cable or a wireless connection.

25. Give three general reasons for the importance of computer networks.

Networks are used to share computer hardware. It reduce costs and make it possible to take advantage of expensive computer equipment. Networks are also used to share data and programs that increases productivity. Networks are also used to communicate with different people all over the world easily.

26. List some benefits of computer networks.

Information & Resource Sharing – Data and information can be shared among different users.

Money Saving – People can save a lot of money by using the same software over a network instead of purchasing separate copy of the software for each user.

Easy Communication – A person can communicate with different people very easily using a large network like Internet.

27. Differentiate between client and serve.

A client computer is connected with a server to access different resources. It sends request to the server for resources. A server computer provides services to the computers and other devices connected to the network. It is more powerful than client.

28. Why server computer is more powerful than client computers?

Server computer provides services to client computers. It performs more activities than client computers such as processing data, controlling access and sharing software etc. That is why the server computer is more powerful.

29. What do you know about the most common type of network?

LAN is the most common type of computer network. It covers a small area. It usually connects the computers and other devices within an office, a building or group of adjacent buildings.

30. Define node.

Each computer or device in the LAN network is called node.

31. How many types of computer networks are available?

There are three types of computer networks. These include local area network, metropolitan area network and wide area network.

32. What is LAN? Give an example.

LAN is the most common type of computer network. It covers a small area. It usually connects the computers and other devices within an office, a building or group of adjacent buildings. The computer lab in a college is an example of LAN.

33. List some uses of LAN.

1. One copy of a software can be shared by all users in a LAN.
2. System resources like printers can be shared between several people in the network.
3. It is easy to manage the data stored on a centralized computer in the network.
4. The data is more secure from being copied or destroyed.
5. Data can be shared by all users using that network.

34. What is peer-to-peer network?

Peer-to-peer LAN is a simple and inexpensive network. It normally connects less than ten computers. Each computer in this network can share hardware, data or information of any computer in the network. Each computer stores files on its own storage devices.

35. What is client-server network?

Client-server LAN is a network in which one or more computers work as servers and other computers work as clients. Server provides services for the clients. It controls access to hardware and software on the network. It provides centralized storage area for progranms, data and information. Client computers request resources from the server.

36. How is peer-to-peer network different from client/server?

In a peer-to-peer network, each computer can communicate directly with other computers on the network. Therefore, each computer plays the role of a server with respect to the other computers that have access to it over the network.

37. How does a file server differ from a database server?

File servers control access to file and disk resources. One problem with file servers is that when users log in to the server to request a file, the server must send the entire file across the network to the user's local machine. In contrast, the database server retrieves the data record, processes the data.

38. When might a peer-to-peer LAN be a good LAN choice?

A peer-to-peer LAN does not require specialized server software, and therefore can be a less costly implementation. For small LANs of generally no more then 15 devices, a peer-to-peer LAN can be a good solution. As the LAN grows, however, a peer-to-peer LAN can become difficult to manage.

39. What is MAN? Give an example.

MAN is a type of network that covers an area of a city. MAN is larger than LAN but smaller than WAN. It is usually used to connect two or more LANs in a city or town. Cable TV network in a city is an example of MAN.

40. What is WAN? Give an example.

WAN covers a large area. It connects computers and other devices in different cities and countries. WAN usually consists of several LANs. The computers in WAN are often connected through telephone lines. The Internet is an example of WAN.

41. Write the most distinctive difference between LAN from WAN.

The most distinctive difference is the distance that is spanned by the network.

42. Compare LAN and WAN transmission speeds.

LAN transmission speeds are typically 10 Mbps to 100 Mbps. In contrast, most WAN communication links operate from 56 kbps to a few megabits per second.

43. How are WANs different from MANs and LANs?

WAN is one of the oldest kinds of data communication networks. WAN typically covers a wide geographical area. It interconnects networks located at geographically distributed sites. Its transmission speed is lower than those for LANs and MANs.

44. Describe the distinguishing characteristics of a WAN, LAN, and MAN. Under what circumstances do companies use each type of network?

The main difference between three is the distance covered. WAN connects sites across states, countries or continents. LAN connects computers and devices within an office or series of offices usually within a few hundred feet to several miles. MAN transmits data and information over citywide distances and at greater speeds than LAN.

45. Do LANs use the same types of communications channels as WANs?

No, LANs and WANs use different communication channels. LANs usually do not use public access networks or satellite transmissions. They primarily use physical media or wiring. WANs use public access networks and satellite to address the large distances across which they send signals

46. What is the purpose of dedicated server?

A dedicated server is used to perform a specific task. File server is used to store and manage files. Print server is used to manage printers and print jobs. Database server is used to manage database.

47. Differentiate among LAN, MAN and WAN.

LANs use direct cabling, wireless radio or infrared signals to connect computers in a small area. MANs use high-speed fiber-optic lines to connect computers located at various places in a major urban region. WANs use long-distance transmission media such as phone lines, microwave or satellites.

48. Define the term network topology.

The physical layout or the way in which network connections are made is called a topology. It refers especially to locations of computers and how the cable runs between them. Bus, Ring and Star are examples of network topologies.

49. What is Bus topology?

Bus topology is the simplest type of network. It supports a small number of computers. In bus topology, all computers are connected to a common communication medium. This medium is often a central wire called bus. It is mostly used in peer-to-peer networks

50. How does star topology work?

All computers in star topology are connected with central device called **hub**. The sending computer sends the data to hub. The hub sends data to the receiving computer. Each computer in star network communicates with a central hub. If hub fails, the whole network become useless.

51. Why star topology is the best topology?

Star toplogy is the best LAN topology. Scalability and reliability of star topology makes it the best topology than others. It is easy to remove or add a device in this topology. It is easier to troubleshoot than other topologies.

52. Distinguish among star, ring and bus topologies.

In star network topology, each computer is directly connected to every other computer on the network. In a ring network topology, all computers form a ring. In a bus network topology, all computers are connected to a common communication medium called bus. A star topology includes a central device or hub to connect all computers together. Ring topology uses token passing and bus topology uses Ethernet.

53. Differentiate between a bus topology and a star topology.

A bus topology usually uses one or more pieces of cable to form a single line or bus. A star topology includes a central device or hub to connect all computers together. Star topology is more expensive to implement than bus topology.

54. What is Ring topology?

In this topology, each computer is connected to the next computer with the last one connected to the first. This topology forms a circle. All computers in ring topology have equal access to the network.

55. Differentiate between bus topology and ring topology.

In a ring network topology, all computers form a ring. In a bus network topology, all computers are connected to a common communication medium called bus. Ring topology is difficult to troubleshoot but bus topology is easy to troubleshoot. The speed of ring topology is higher than bus topology.

56. Define standard.

The standards are the documents that contain technical and physical specifications about the network being designed. The networks can be reliable and efficient by following certain standards.

57. Describe De Facto standard.

De facto means by tradition or by facts. These standards are developed without any formal planning. These standards come into existence due to historical developments. These standards are still being used by the organizations all over the world.

58. Describe De Jure standard.

De Jure means according to law or regulation. These are the standards which have been properly approved by networkings governing body. These are developed with proper research and design to fulfill the requirement of data communication.

59. Distinguish between de facto and de jure standards.

De facto standards are developed without any formal planning. De Jure standards are properly approved by networkings governing body. These are developed with research and design to fulfill the requirement of data communication. Many de facto standards become de jure standards when analyzed and improved by industry standards associations.

60. List the organizations that develop protocols and standards.

CCITT – It is United Nations agency. It is responsible for defining standards for telephone, telegraph and data communication.

IEEE – It also defines data communication protocols. Many communication protocols used in Local Area Network are defined by IEEE:

ISO – It is responsible for defining standards in various fields.

61. List the basic functions of communication protocols.

The main functions of communication software or protocols are as follows:

Data Sequencing – It is a process of breaking a long message into smaller blocks.

Data Routing – It is a process of finding the most efficient path between source and destination.

Data Flow – It is a process that controls data transmission properly if the sender computer is faster than the receiver computer.

Error Control – It ensures transmission of data without error and solves the problem if an error occurs.

62. Write the names of different LAN protocols.

Different LAN protocols are Ethernet, Token ring and ARCnet.

63. What is the use of Ethernet?

Ethernet is a LAN technology. It is based on bus topology but Ethernet network can be wired in star topology also. It is the most popular LAN because it is inexpensive and easy to install and manage.

64. What is the use of token ring?

Token ring is also a LAN technology. It allows network devices to access the network by passing a special signal called **token**. Token is like a ticket. A device can transmit data over the network only if it has a token.

65. What is ARCnet?

ARCnet stands for **Attached Resource Computer Network**. It is both a topology and networking technology. It uses twisted-pair or coaxial cable. Original ARCnet protocol was slow. It became popular as it was inexpensive, reliable and easy to setup and expand.

66. What is packet switching?

When a computer sends data over the Internet, data is divided into small pieces called packets. Each packet contains data and destination, sender and the sequence information used to reassemble the data at the destination. This process is known as packet switching.

67. What is a key characteristic of a circuit-switching service?

A key characteristic of a circuit-switching service is that it creates a direct connection between two communicating devices. The term "path" is used for circuit-switching services between the two communication devices. While two devices use circuit path, no other devices will be able to use circuit.

68. What is the relationship between circuit and circuit configuration?

A circuit is the link that provides the physical means for data transfer. Circuit configuration affects the way in which two or more communicating devices share their link.

69. How does circuit switching differ from packing switching?

Circuit switching creates a direct connection or path between two devices. The term path is used with circuit switching and the term link is used with packet switching. When two devices establish a path using a circuit-switched connection, their link will be at one constant rate. The rate will be determined by the slower device.

70. Distinguish between frames and packets.

Frames are messages in single networks. Packets are messages that are sent through the Internet. In each network, the packet is carried in a frame limited to that network.

71. What is a communication device?

The hardware used to transmit data, instructions and information between a sending and receiving device is called communication device.

72. What is NIC?

Network card is a communication device used to connect a device to a network. It is also called **network interface card (NIC)**. It manages the transmission of data, instructions and information to and from the device.

73. What is MAC address?

The NIC contains a unique network address called **MAC address**. MAC stands for **media access control**. It is 6 bytes in length and is displayed in 12 hexadecimal digits. The first 6 digits are used to indicate the vendor of the network interface also called **organizationally unique identifier**. The last 6 numbers form a unique value for each NIC assigned by vendor.

74. How does a modem allow computers to communicate over telephone lines?

Modem translates computer data into signals compatible with the telephone system.

75. Define modulation. Why is it necessary?

The process of converting digital signal into analog signal is called modulation. Computer stores data in digital form. Since a modem transmits data using a telephone line, so it is converted from digital to analog form.

76. Define demodulation. Why is it necessary?

The process of converting analog signal into digital signal is called demodulation. The modem on receiving computer receives data in analog form. The incoming analog data is converted back into digital format to be used by the computer.

77. Define downstream rate and upstream rate.

The rate of receiving data is called downstream rate and the rate of sending data is called upstream rate.

78. What is network bridge?

A **network bridge** can be used in computer networks to interconnect two LANs and to separate network segments. The bridge is a layer 2 device in the OSI model It uses MAC address information for making decisions to forward data packets. It only forwards the data that needs to be sent across the bridge to the adjacent network segment. This makes it possible to segment the network data traffic.

79. What is the difference between Hub and Switch?

Hub broadcasts all messages to every computer on the network but only the intended recipient computer takes the message. Network switch is more intelligent device than hub. The switch improves the efficiency of the data transfer in the network. It receives a message from any device connected to it and then transmits the message only to that device for which the message was meant.

80. How does a layer 3 switch differ from a layer 2 switch?

Layer 3 switch still works at layer 2 but additionally works at the layer 3 of the OSI model. The major difference is that the packet switching in basic routers is handled by a programmed microprocessor. The layer 3 switch uses application-specific integrated circuits (ASICs) hardware to handle the packet switching. The advantage of using hardware to handle the packet switching is a significant reduction in processing time.

81. What is router?

Router connects multiple networks using similar or different protocols. It manages the best route between two communication networks. Routers are used when several networks are connected together. The router is a layer 3 device in OSI model. It means that it uses the network address to make routing decisions to forwar data packets.

82. What is communications channel?

The path through which data is transmitted from one place to another is called communications channel. It is also known as communication media or transmission media.

83. How media is important in communication?

Communication network cannot exist without a medium. The medium is used to connect the source and receiver and provide path to send messages. A wide variety of transmission media can be used in a communication network.

84. What is bandwidth?

The amount of data that can be transferred through a communication medium in a unit of time is called bandwidth. The bandwidth of digital signal is measured in bits per second or Bytes per second. The bandwidth of analog signals is measured in cycles/Seconds or Hertz.

85. What is bandwidth requirement?

Each type of digital data needs minimum data transfer rate. It is called bandwidth requirements.

86. Distinguish between the bit rate and the baud rate.

The **bit rate** is the rate at which data is transmitted. The **baud rate** is the number of clock cycles that a transmission system uses per second.

87. What is difference between guided and unguided transmission media?

The guided transmission media are used to connect communication devices directly with each other using different types of cables. Coaxial cable and fiber optics are examples of guided media. The unguided media are used to connected communication devices through air or space using broadcast radio signals etc. Satellite is an example of unguided media.

88. Name some guided media.

The guided media include twisted pair, coaxial cable and fiber optics.

89. What do you mean by physical transmission media?

The physical transmission media are used to connect communication devices directly with each other using different types of cables. It is also called bounded media or guided media.

90. Name different types of transmission media.

Different types of transmission media are guided transmission media and unguided transmission media. Twisted pair cable and coaxial cable are guided media. Satellite communication and Bluetooth are unguided media.

91. What is the use of twisted pair?

Twisted pair is the most commonly used physical transmission medium. It is used in local area network to connect computers and other devices.

92. Difference between UTP and STP.

Shielded twisted pair (STP) contains a layer of foil shielding. The shielding reduces the electromagnetic interference. Unshielded twisted pair (UTP) does not contain a layer of foil shielding. STP cable is more expensive than UTP.

93. What is coaxial cable?

Coaxial cable consists of copper wire covered by an insulating material. The insulated copper wire is covered by copper mesh. The mesh protects the data signals from interference by external electromagnetic waves.

94. Why fiber optic cable is very popular?

Fiber optic cable is very popular because it is the fastest communication medium. It is not affected by electromagnetic waves.

95. Describe the roles of core and cladding in optical fiber.

The core carries the light signal. Cladding reflects the signal back into the core with perfect internal reflection so that no light escapes from the core.

96. Write two advantages and two disadvantages of fiber optics cable.

The advantages are that it provides faster data transmission and better security for signal during transmission. The disadvantages are that it is difficult to install and very costly.

97. What are the alternatives to wire or fiber-optic cable?

There are several alternatives to wire or fiber-optic cable as the transmission media. However, these options have not got much acceptance. The technologies have not been fully developed and they are expensive. Some examples of wireless transmission media include Infrared, Laser and Radio.

98. What is cellular communication? How it is helpful in daily life?

Cellular communication is a wireless communication system. It is widely used all over the world for mobile communication. It is used to transmit text, images, voice and video. It is helpful in daily life as it provides the facility to communicate easily through mobiles from anywhere in the world.

99. How does cellular communication work?

Cellular communication divides a physical region into different sections known as cells. Each cell has a low-powered radio transceiver known as base station or cell site. These cells are linked together to enable a large number of cellular phones to communicate with one another.

100. How does satellite communication work?

Satellite communication is used to communicate over large distances. It uses communication satellites and earth-based stations. A satellite receives microwave signals from earth-based station. It amplifies the signals and retransmits them back to different earth-based station.

101. Write one advantage and one disadvantage of satellite communication.

The advantage is that it can be used to communicate over a large distance. The disadvantage is the high cost to put the satellite around globe.

102. Write some important applications of satellite communication work.

The satellite communication is used for international telephone calls, weather forecasting and television broadcasts. It provides communication to remote areas and to ships and aircrafts etc.

103. What is microwave system?

Microwaves are radio waves that are used to provide high-speed transmission. Both voice and data can be transmitted through microwave. Data is transmitted through the air from one microwave station to another similar to radio signals.

104. How does microwave system work?

Microwave uses line-of-sight transmission. It means that the signals travel in a straight path and cannot bend. Microwave stations are placed within 20 to 30 miles to each other. Each station receives signal from previous station and transfer to the next station.

105. What is the disadvantage of Microwave System?

A disadvantage of microwave is that it is limited to line-of-sight transmission. This means that Microwave signals must be transmitted in a straight line. There can be no obstruction such as buildings or mountains, between microwave stations.

106. Discuss any three communication media in terms of security, cost and speed.

Security: Fiber optic cable is the most secure transmission medium. Fiber optic is harder to tap. Twisted pair and coaxial cable are easily intercepted.

Cost: Twisted Pair cable is the least expensive media for data communications. Coaxial cable is more expensive than twisted pair but less expensive than fiber optic. The most expensive form of wiring is fiber optic cable.

Speed: Fiber optic cable has the highest possible data transmission speed. Data can be transmitted up to 10 Gbps. Coaxial cable can support data transmission rates higher than a twisted pair cable. It has a data transfer rate of 10 Mbps.

107. Explain some of the performance differences between twisted pair, coaxial cable and fiber optic cable.

A twisted pair cable is the slowest medium for data transmission. Coaxial cable suffers from little distortion from external signals because it consists of a copper data transmission wire surrounded by insulation. It can support data transmission rates higher than a twisted pair cable. A fiber optic cable carries optical signals that are transmitted at the speed of light. It can support the highest data transmission rate required for broadband networks.

108. Describe four types of cable-less communications media.

The four types of cable-less communication media are microwave, satellite, infrared, and radio waves. Microwave uses high frequency radio signals to send data through the air. Satellite uses high frequency radio signals from microwave stations to a communications satellite. Infrared transmits data and information in coded form by means of an infrared light. Radio waves can be used to transmit data. The frequencies used are rented from public radio networks.

109. What is transmission impairment?

The signals travel through transmission media. The transmission media are not perfect and cause **signal impairment**. It means that the signal at the beginning of the medium is not the same as the signal at the end of the medium.

110. What is Attenuation?

Attenuation means a loss of energy. A signal loses some of its energy in overcoming the resistance of the medium when it travels. That is why a wire carrying electric signals gets warm after some time. Some of the electrical energy in the signal is converted into heat.

111. What is noise?

Noise is the electrical disturbance that can degrade the communications.

112. What are software protocols?

A network protocol that determines how data is transferred from one place to another is called software network protocol. It is used to encode and decode data during transmission over a computer network.

113. What are hardware protocols?

A network protocol that determines how hardware devices communicate with one another is called hardware network protocol. It is used to determine the voltage levels and types of wires used for data communication.

114. What are the two dominant network standards architectures?

Two dominant network standards architectures are **OSI** and **TCP/IP.**

115. What is an IP Address?

An IP address is a unique 32-bit address that identifies on which network the computer is located as well as differentiates the computer from all other devices on the same network.

116. How IP addresses are classified?

IP addresses are classified as IPv4 or IPv6. IP version 4 (IPv4) is the current TCP/IP addressing technique being used on the Internet. The address space for IPv4 is quickly running out due to the rapid growth of the Internet and the development of new Internet-compatible technologies.

117. How is IPv6 better than IPv4?

The **Internet Protocol version 6 (IPv6)** is better than IPv4 in that IPv6 increases the size of IP address fields from 32 bits to 128 bits. It provides a large number of IP addresses (2^{128}).

118. What is IANA?

The agency that assigns IP addresses to computer networks and makes sure no two different networks are assigned the same IP network address.

119. Write a short note on OSI model.

OSI stands for **Open System Interconnection. An open system is a model that allows any two** systems to communicate even if their architectures are different. It covers all aspects of network communications.

120. Why was the OSI model developed?

The purpose of OSI model is to give software developers an alternative to the proprietary protocols developed by computer manufactures. OSI model was designed to provide a standard way for computers to communicate so that different computers can be used on the same network.

121. List different layers of OSI model.

OSI model has seven layers. These are Application layer, Presentation layer, Session layer, Transport layer, Network layer, Data link layer & Physical layer.

122. Identify the OSI level at which each of the following operates.

- Bridge - data link layer of the OSI model
- Repeater - physical layer
- Router - network layer
- Gateway - transport layer or above

123. What is Point-to-Point protocol?

The **Point-to-Point Protocol** (PPP) is an important standard at this OSI level. It is used for communications across point-to-point links. It is an important protocol for wide area networking.

124. What is Network layer?

The network layer is responsible to establish, maintain, and terminate network connections. It manages the delivery of data from source to destination.

125. What is Transport layer?

The transport layer controls the flow of data. It ensures that messages are delivered error free. It divides large messages into small packets for efficient transmission.

126. What is Session layer?

The session layer establishes, manages, and terminates user connections. A session is an exchange of messages between computers. It synchronizes user tasks.

127. List and explain some functions provided by Physical Layer

- **Characteristics of Media:** The physical layer defines the characteristics and type of transmission medium.
- **Representation of Bits:** The bits are encoded into electrical signals for transmission. The physical layer defines the type of encoding.
- **Data Rate:** Physical layer defines the number of bits that will be sent in each second.
- **Synchronization:** The clocks of sender and receiver are also synchronized.
- **Line Configuration:** The physical layer defines the attachment of communication devices with medium.
- **Transmission Mode:** The physical layer defines the direction of transmission between two devices: simplex, half-duplex or full-duplex.

128. List and explain some functions provided by Data link Layer

- **Framing:** Data link layer divides the stream of bits into manageable data units called frames.
- **Flow Control:** If the rate of sending data is more than the rate of receiving data, data link layer imposes a flow control mechanism to prevent this situation.
- **Error Control:** The data link layer detects and retransmits damaged or lost frames. It also prevents the duplication of frames.
- **Access Control:** If two or more devices are connected to the same link, data link layer determines which device has control over the link at a given time.

129. List and explain some functions provided by Presentation Layer.

- **Data Reformatting:** When two computers exchange data, the data is changed to bit streams before it is transmitted. Two computers may use different encoding techniques. The presentation layer at sending computer changes data according to the sender's format. The presentation layer at receiving computer changes data according to the receiver's format.
- **Encryption:** The presentation layer encrypts the data before transmission. It means that the sender transforms the original information to another form and sends the resulting message over the network. The receiver again transforms the message back to its original form. It is called decryption.
- **Compression:** Data compression reduces the number of bits to be transmitted. The presentation layer compresses a large amount of data into small size.

130. What is an Application layer?

The application layer is the top-most layer of OSI model. It provides services directly to user applications. It enables the user to access the network.

131. What errors does the transport layer usually fix?

The transport layer usually fixes all errors created at the transport layer or lower layers.

132. Which layers are network support layers?

a. Physical Layer
b. Data link Layer and
c. Network Layers

133. Which layers are user support layers?

a. Session Layer
b. Presentation Layer and
c. Application Layer

134. Which layer links the network support layers and user support layers?

The Transport layer links the network support layers and user support layers.

135. What do application layer standards govern?

Application layer standards govern how two applications work with each other, even if they are from different vendors.

136. List the five layers of the TCP/IP protocol suite.

From highest to lowest these are: Application, Transport (Host-to-Host), Internet, Network Access and Physical

137. List key protocols found application, transport, and Internet layers.

- Application layer protocols include: FTP, HTTP, SMTP, SNMP, and Telnet
- Transport layer protocols include: TCP and UDP
- Internet layer protocols include: IP, RIP, and RTP

138. What is the role of IP in the TCP/IP protocol suite?

IP is used for:

- Transmitting data from higher-level protocols like TCP, UDP in IP datagrams, from one host to another host in the network.
- Identifying individual hosts in a network using an IP address.
- Routing datagrams through gateways and
- Fragmenting and reassembling datagrams based on the MTU of the underlying network.

139. How does TCP differ from UDP?

TCP is a connection-oriented protocol. It means that the sender cannot send until the receiver is contacted and agrees to a communication. UDP is a connectionless service. It means that the sender does not have to first establish a link to the receiver before beginning to transmit data.

140. What is the purpose of TCP sequence numbers?

The purpose of TCP sequence numbers is to allow the receiving transport process to put arriving TCP segments in order if IP delivers them out of order.

141. Why should we prefer UDP over TCP?

Depending on the data being transmitted, UDP can be the better choice than TCP. UDP does not require the sender and receiver to establish a connection before sending data. It saves time and increases efficiency.

142. How are OSI and TCP/IP models similar?

Both models have several characteristics in common. Both are open architecture models and are based on a layered architecture. Each model can be broken into several distinct components called layers. Each model is well established and accepted by the data and telecommunications industries as models.

143. What are the data units at different layers of TCP / IP protocol suite?

The data unit created at the application layer is called a message. The data unit created at the transport layer is called segment or user datagram. The data unit created at the network layer is called the datagram. The datagram is encapsulated into a frame at the data link layer. It is finally transmitted as signals along the transmission media.

144. What is difference between ARP and RARP?

ARP is used to associate the 32 bit IP address with the 48 bit physical address, used by a host or a router to find the physical address of another host on its network by sending a ARP query packet that includes the IP address of the receiver. RARP allows a host to discover its Internet address when it knows only its physical address.

145. What is the difference between TFTP and FTP application layer protocols?

TFTP allows a local host to obtain files from a remote host but does not provide reliability or security. It uses the fundamental packet delivery services offered by UDP. FTP is the standard mechanism provided by TCP / IP for copying a file from one host to another. It uses the services offered by TCP and so is reliable and secure.

146. What is ICMP?

ICMP is a network layer protocol of the TCP/IP suite used by hosts and gateways to send notification of datagram problems back to the sender. It uses the echo test / reply to test whether a destination is reachable and responding. It also handles both control and error messages.

147. Define the terms Unicasting, Multiccasting and Broadcasting.

If the message is sent from a source to a single destination node, it is called unicasting. If the message is sent to some subset of other nodes, it is called uulticasting. If the message is sent to all the nodes in the network it is called broadcasting.

148. What is encapsulation?

Encapsulation is placing a message in data field of another message

149. What layers govern transmission in LANs and WANs?

The physical and data link layers govern transmission in single LANs or WANs.

150. What are the benefits provided by a bridge?

- Expand the length of an existing network
- Increase the number of workstations on the network
- Reduce traffic congestion (network partitioning)
- Provide a connection to a dissimilar network (e.g., Ethernet to Token Ring)
- Move data across a intermediate network with a dissimilar protocol

151. Describe the conditions under which a server-based network is recommended over a peer-to-peer network.

- A large number of workstations need to access a particular resource.
- Security is a significant concern.
- Adequate administration support is available.
- Projected future growth of the network is planned.
- A heterogeneous environment exists (i.e., Macs and PCs).
- Users are widely dispersed.
- Individual workstations lack the necessary computing power or resources to support a peer-to-peer network

152. What is the use of firewall?

Firewall is a set of components that are used to restrict the access to data and information on a network. It may consist of hardware and software. Firewalls are used to restrict the unauthorized users from accessing the resources of an organization.

Multiple Choice

1. **A collection of computers connected together is called:**
   a. Processing b. Network c. Chatting d. Centralized system
2. **The technology of long-distance communication is known as:**
   a. Mass communication b. Telephony c. Telecommunications d. None
3. **Which of the following is NOT a benefit of computer networks?**
   a. Reduce hardware costs b. Connect people
   c. Enable shared applications d. Produce high quality programs-

4. Each computer on a network is called a:
a. Bus b. Terminator c. Node d. None

5. Companies that use networks can save time and money because networks allow users to share:
a. Hardware devices b. Software programs c. Information d. All

6. Many networks include a central computer that may be called:
a. Server b. Bridges c. Gateways d. Client

7. A device used to connect two computers lying in same office or building is called:
a. Ethernet Card b. Graphics Card c. Modem d. Sound Card

8. Which of the following is not a category of network?
a. WAN b. LAN c. MAN d. NAN.

9. LAN stands for:
a. Local area nodes. b. Logical arrangement of networks.
c. Local area network. d. Linked-area network.

10. A network that covers a limited geographic distance such as an office is called:
a. Centralized network b. Metropolitan area network
c. Local area network d. Wide area network

11. In a traditional LAN, each computer on the network is connected through:
a. Cables. b. Satellites. c. Microwaves. d. Wireless transmission.

12. Which of the following is a component of LAN?
a. Cable b. NIC c. Bridge d. All

13. LAN is a combination of
a. Network adapter cards b. LAN cables c. LAN Application software d. All

14. Which problem occurs when two workstations on shared Ethernet try to access LAN at the same time?
a. Termination b. Deadlock c. Collision d. Concession

15. A network that covers a large geographic distance such as a country is called a:
a. Centralized network. b. Distributed network.
c. Local area network. d. Wide area network

16. The primary difference between a LAN and a WAN is:
a. The number of software programs available. b. Distance.
c. The variety of hardware devices. d. The number of hardware devices.

17. What type of network is the Internet?
a. LAN b. MAN c. WAN d. None

18. MAN stands for:
a. Metropolitan area network b. Marked area network
c. Metropolitan arranged network d. Manufactured arrangement of networks

19. A network that transmits data over citywide distances faster than LAN is:
a. WAN b. LAN c. MAN d. NAN

20. NIC stands for:
a. Network internal card. b. Newer industrial computer.
c. Networking Internet connection. d. Network interface card

21. NIC allows direct connection to:
a. Monitor b. A network c. A printer d. A modem

22. A device that is used to connect two computers via an ordinary telephone line is:
a. Ethernet Card b. Graphics Card c. Modem d. Sound Card

23. Which of the following is not a LAN topology?
a. Bus b. Band c. Star d. Ring

24. Which is the most common LAN topology for microcomputer LANs?
a. Bus b. Band c. Star d. Ring

25. The physical layout of a LAN is known as:
a. Topology b. Session c. Link d. Connector

26. One or more computers connected to hub computer is called:
a. Ring Network b. Band c. Bus d. Star Network

27. Topology that is used for a small number of computers is called:
a. Star b Bus c. Mesh d. Ring

28. A connection for similar networks uses:
a. Bridgeb. Gateway c. Both d. None

29. The set of rules to exchange data in a communications network is called:
a. Gateway b. Procedure c. Protocol d. Token

30. A device used to connect dissimilar networks:
a. Gateway b. Bridge c. Both a and b d. None

31. bps stands for:
a. Bits per second b. Binary per second c. Bytes per second d. Baud per second

32. People on LAN can share:
a. Printer b. Modem c. CD-ROM/disk Drive d. All

33. Software to peruse the internet is called:
a. Gateway b. EFT c. Browser d. Teleconferencing

34. ISDN stands for:
a. Internet Service Digital Network b. Improved Speed Digital Network
c. Integrated Services Digital Network d. Immediate Synchronous Digital Network

35. DSL stands for:
a. Direct Service Lease b. Domain Server Link
c. Distant Service Line d. Digital Subscriber Line

36. Which of the following is NOT a network communications device?
a. Router b. Hub c. LAN d. NIC

37. The process of transferring data electronically from one place to another is called:
a. Data processing b. Data Communicationc. Data sequencing d. Data Sender

38. The electromagnetic or light waves representing data are called:
a. Information b. Signal c. Sender d. None

39. The number of times a wave repeats during a specific time interval is called:
a. Pulse b. Amplitude c. Frequency d Oscillation

40. The height of wave within a given period of time is known as:
a. Frequency b. Amplitude c. Oscillation d. Pulse

41. Transmission permitting data to move only one way at a time is called:
a. Half-duplex b. Simplex c. Full-duplex d. Start/stop

42. An arrangement in which data can be received and sent simultaneously is called:
a. Simples b. Full-duplex c. Half-duplex d. Multi-duplex

43. A telephone conversation is an example of:
a. Full-duplex transmission b. Half-duplex transmission.
c. Simplex transmission. d. Asynchronous transmission.

44. Television and radio broadcasts are examples of:
a. Full-duplex transmission b. Half-duplex transmission
c. Simplex transmission d. None

45. Internet surfing is an example of:
a. Simplex b. Half duplex c. Full duplex d. None

46. Which transmission allows data to travel in both directions but only one direction at a time?
a. Simplex b. Half Duplex c. Full Duplex d. Reverse Duplex

47. Full-duplex communication is made possible by devices called:
a. Multiplexer b. Modem c. Keyboard d. Mouse

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

48. Which of the following is the fastest communication mode?
a. Half duplex b. Full duplex c. Simple d. None

49. Analog signal is measured in:
a. Volt b. Hertz c. Digits d. WATTS

50. Sound, light and radio waves are examples of:
a. Digital signal b. Analog signal c. Simple signals d. None

51. Which of the following consists of thin glass to transmit beams of light?
a. Twisted pair b. Coaxial cable c. Fiber-optic cable d. None

52. Which of the following transmits voice and data through air as high-frequency radio waves?
a. Twisted pair b. Coaxial cable c. Fiber-optic cable d. Microwave

53. All of the following are guided communications media EXCEPT:
a. Twisted pair b. Fiber-optic cables c. Coaxial cables d. Satellite

54. Which of the following is not a communication media?
a. Twisted Pair b. UTP c. Microwave d. Modem

55. Modem stands for:
a. Modification/Demodification b. Modulation/Demodulation
c. Mode/Modeless d. None

56. Which is the correct measurement of a modem's data transfer rate?
a. Kbps b. Gbps c. bps d. Mbps

57. A communications signal in the form of a continuous wave is called:
a. Digital. b. Modulation. c. Analog d. None

58. The process of converting from analog to digital signal is known as:
a. Modulation b. Data routing c. Data sequencing d. Demodulation

59. Converting a digital signal to an analog signal is called:
a. Modulation b. Demodulation c. Conversion d. None

60. Signals produced by a computer to send over phone line must be converted to:
a. Modems b. Analog signals c. Digital signals d. Microwaves

61. A modem's rating of 56K refers to its:
a. Memory size b. Transmission speed c. Modem Size d. None.

62. A modem:
a. Derives its name from modulator-demodulator
b. Converts digital signals into analog signals
c. Converts analog signals into digital signals
d. All

63. Which of the following features is provided with a modem?
a. Speeds b. Self-testing c. Transmission rate d. All of these

64. Bps is short for:
a. Baud per second b. bytes per second c. Bits per second d. binary packets a second

65. Communication between a computer and keyboard involves:
a. simplex b. Half-duplex c. Full-duplex d. All

66. A network configuration that allows every computer to act as client and as well as server is:
a. Client/server model b. Host system
c. Server network d. Peer-to-peer network

67. Which transmission media is least susceptible to noise?
a. Microwave b. Fiber optics c. Twisted pair d. None

68. Which communications medium has the highest bandwidth?
a. Fiber optic b. Coaxial c. Copper d. Twisted pair

69. An arrangement in which most of the processing is done by server is called:
a. Client/server model b. Peer-to-peer
c. Simplex transmission d. File server

70. Which method places all processing, hardware and software in one location?
a. Distributed data processing b. Centralized data processing
c. Parallel Data processing d. None

71. Which processing method allows both remote access and remote processing?
a. Distributed data processing b. Centralized data processing
c. Digital Transmission d. None

72. The protocol that makes Internet universality possible is called:
a. TCP/IP b. Token Ring c. Ethernet d. None

73. OSI Reference Model has:
a. 7 layers b. 3 layers c.6 layers d. 1 layers

74. Which layer is responsible for establishing, maintaining and terminating a user connection?
a. Presentation b. Application c. Transport d. Session

75. Which layer is concerned with addressing and routing?
a. Network layer b. Data Link Layer c. Physical Layer d. Transport Layer

76. Data link have two separate sub-layers:
a. LLC and MAC b. CS and MA c. Token and Ring d. Ethernet

77. The top most layer of the OSI is:
a. Application b. Session c. Transport d. Presentation

78. The bottom layer of OSI Model is:
a. Application b. Physical c. Transport d. Presentation

79. A layer that controls how bit-streams of data is sent and received over physical medium is:
a. Application b. Session c. Transport d. Physical

80. A layer responsible for flow control and ensuring messages are delivered without error is:
a. Application b. Session c. Transport d. Physical

81. A layer that defines formats the data uses a it in transmitted on the communication line is:
a. Application b. Presentation c. Transport d. Physical

82. The layer that provides services directly support user application:
a. Application b. presentation c. Transport d. Physical

83. OSI stands for:
a. Open system Interconnection b. Open system international
c. Open Small Internet d. Open system Interlink

84. What layer of the OSI models does data compression?
a. Network b. Data Link c. Presentation d. Physical

85. Switches and cables are examples of:
a. Protocols. b. Standards. c. Routers. d. Physical media.

86. Two types of LAN are Client/Server and:
a. P2PN. b. NIC. c. NOS. d. WAN.

87. _______________allows people to communicate face to face over long distances by combining video and computer technology.
a. Telephony b. Television c. Video beaming d. Video teleconferencing

88. Which form of telecommunication acts like a remote printer for a document?
a. File server b. Fax machine c. Router d. Fax modem

89. Bridges can not be the solution to
a. Limit distance b. Limit traffic
c. Limite number of stations d. Paket redundancy

90. In a ________ connection, three or more devices share a link.
a. Multipoint b. Point-to-point c. a and b d. None

91. In a ______ connection, two and only two devices are connected by a dedicated link.
a. Multipoint b. Point-to-point c. a and b d. None

92. Frequency of failure and network recovery time after a failure are measures of the _______ of a network.

a. Performance b. Reliability- c. Security d. Feasibility

93. An unauthorized user is a network _______ issue.

a. Performance b. Reliability c. Security d. All

94. Which agency developed standards for physical connection interfaces and electronic signaling specifications?

a. EIA- b. ITU-T c. ANSI d. ISO

95. _______ refers to the structure or format of the data, meaning the order in which they are presented.

a. Semantics b. Syntax c. Timing d. All

96. ________ defines how a particular pattern to be interpreted, and what action is to be taken based on that interpretation.

a. Semantics b. Syntax c. Timing d. None

97. IPv4 address is:

a. 8 bit b. 16 bit c. 32 bit d. 64 bit

98. DHCP is a abbreviation of:

a. Dynamic Host control b. Dynamic Host Configuration Protocol
c. Dynamic Hyper control protocol d. Dynamic Hyper Configuration Protocol

99. Why was the OSI developed ?

a. Manufacturers disliked the TCP/IP protocol
b. The rate of data transfer was increasing exponentially
c. Standards were needed to allow any two systems to communicate
d. None of these

100. Telnet, FTP, SMTP, DNS, HTTP are examples of protocols that are used in __________.

a. Application layer of OSI reference model
b. Presentation layer of OSI reference model
c. Session layer of OSI reference model
d. Data link layer of OSI reference model

101. Routers operate at which layer of the OSI Model?

a. Physical b. Transport c. Network - d. MAC sublayer of the Data Link layer

102. Which of the following are Transport layer protocols?

a. TCP & UDP b. ATM c. CISC d. HTTP & FTP

103. Which of the following are considered to be the upper layer protocols?

a. Presentation & Session
b. Application & Presentation
c. Application, Presentation, & Session
d. Application, Presentation, Session, & Transport

104. Flow control takes place at what layer?

a. Physical b. Network c. Transport d. Data Link

105. Encryption takes place at which layer?

a. Physical b. Presentation c. Application d. Session

106. IP is implemented at which OSI Model layer?

a. Transport b. Network c. Data Link d. Presentation

107. Which layer handles the formatting of application data so that it will be readable by the destination system?

a. Application b. Presentation c. Transport d. Network

108. Packets are found at which layer of the OSI Model?

a. Data Link b. Transport c. Network d. Presentation

109. Which layer translates between the physical (MAC) & logical address?
a. Network b. Data Link c. Transport d. Presentation

110. Repeaters & hubs operate at whaich layer?
a. Network b. Physical c. Transport d. Data Link

111. Bit synchronization is handled at which layer of the OSI Model?
a. Physical b. Session c. Network d. Data Link

112. Bridges operate at which layer of the OSI Model?
a. Physical b. Data Link c. Network d. Transport

Answers

1. b	2. c	3. d	4. c	5. d	6. a	7. a
8. d	9. c	10. c	11. a	12. d	13. d	14. c
15. d	16. b	17. c	18. a	19. c	20. d	21. b
22. c	23. b	24. c	25. a	26. d	27. b	28. a
29. c	30. a	31. a	32. d	33. c	34. c	35. d
36. c	37. b	38. b	39. c	40. b	41. a	42. b
43. a	44. c	45. b	46. b	47. a	48. b	49. a
50. b	51. c	52. d	53. d	54. d	55. b	56. c
57. c	58. d	59. a	60. b	61. b	62. d	63. d
64. b	65. a	66. d	67. b	68. a	69. a	70. b
71. a	72. a	73. a	74. d	75. a	76. a	77. a
78. b	79. d	80. c	81. b	82. a	83. a	84. c
85. d	86. a	87. d	88. b	89. d	90. a	91. b
92. b	93. c	94. a	95. b	96. a	97. c	98. b
99. c	100. a	101. c	102. a	103. c	104. c	105. b
106. b	107. b	108. b	109. a	110. b	111. b	112. c

Fill in the Blanks

1. ____________ is the software that enables us to communicate with other system.
2. The procedure of data transmission in the form of software is commonly called ____________.
3. Breaking a long transmission into smaller blocks is called ____________.
4. ____________ is the process of finding the most efficient path between source and destination.
5. In ____________ mode the communication can take place in one direction.
6. In ____________ mode the flow of information is unidirectional.
7. In ____________ mode, data can flow in both direction but not at the same time.
8. Data in ____________ can travel in both directions simultaneous.
9. ____________ is the transmission of data in a continuous wave form.
10. Analog signal is measured in ____________.
11. In ____________ mode, Direction of flow never changes.
12. A combination of four bits is called ____________.
13. Data transmission through a medium can be either synchronous or ____________.
14. In ____________ transmission data is transmitted character by character.

15. In ____________ transmission data is transmitted block by block.
16. In ____________ communication, data is transferred using physical medias like wires.
17. In computer networking area, two types of transmission media are ________ and ________.
18. Wire pairs, coaxial cables and Fiber optics are examples of ____________ media.
19. The transmission from earth station to satellite is called ____________.
20. The transmission from satellite to earth station is called ____________.
21. Microwave signals must be transmitted in a ____________.
22. A ____________ is a microwave station placed in outer space.
23. ____________ is such a satellite that can be accessible from anywhere in Pakistan.
24. In ______, microwave signal is transmitted from a transmitter on earth to the satellite at space.
25. Earth's atmosphere and outer space are examples of ____________ media.
26. The technique by which a digital signal is converted to analog form is known as ________.
27. The technique by which analog signal is converted to digital form is known as ________.
28. The connection between modem and computer is called ____________.
29. DTE stands for ____________.
30. DCE stands for ____________.
31. MODEM stands for ____________
32. ____________ is used to share data & information and communicate with one another.
33. A ____________ connects two computers via an ordinary telephone line
34. A ____________ connects two computers, which are lying in the same building.
35. ____________ is the physical layout of a network.
36. A network must have at least ____________ computers.
37. There are ____________ types of network topologies.
38. Three basic topologies are ____________, ____________ & ____________.
39. All routing is done through the central node or Hub in ____________ topology.
40. The set of rules for exchanging data between computers connected to network is called ________.
41. In ____________ topology, each node is connected to two adjacent nodes or neighbors.
42. In ____________ topology, all nodes are connected to a common communication medium.
43. The interconnecting of a number of computers connected together through one topology is usually called a ____________.
44. In ____________ topology each computer is connected to the next computer with the last one connected to the first.
45. X25 used in ____________ network
46. ISO stands for ____________.
47. ____________ protocols are rules followed by programs to communicate with each other.
48. ____________ Protocol define how hardware devices operate and work together.
49. Ethernet is a ____________ protocol.
50. IBM stands for ____________.
51. ____________ is a device used to join same types of network and boost the signal in a network.
52. A ____________ is a device used to join similar types of networks.
53. A ____________ is a device used to join different types of network and use different protocol.
54. A ____________ is a device used for interconnecting different types of networks together.
55. UTP is the abbreviation of ____________.

56. Two protocols are used in bus topology are ________ and ________.
57. Mbps stands for ________.
58. A network of computers connected through Ether cards is called ________.
59. The amount of data can be transferred through a communication medium in a unit of time is called ________.
60. The loss of strength of signal in a communication media is called________.
61. LAN stands for ________.
62. The cable joining the hubs is called a ________.
63. A network of networks is called________.

Answers

1. Data Communication Software	2. Protocol	3. Data Sequencing
4. Data Routing	5. Simplex	6. Simplex
7. Half	8. Full duplex mode	9. Analog
10. Volts	11. Simplex	12. Nibble
13. Asynchronous	14. Asynchronous	15. Synchronous
16. Bounded	17. Bounded, Unbounded	18. Bounded
19. Uplink	20. Downlink	21. Straight line
22. Communication Satellite	23. INTELSAT	24. Satellite Communication
25. Unbounded	26. Modulation	27. Demodulation
28. Digital Connection	29. Data Terminal Equipment	30. Data Communication Equipment
31. Modulation/Demodulation	32. Computer Network	33. Modem
34. Ethernet Card	35. Topology	36. TWO
37. 3	38. Star, Bus, Ring Topology	39. Star
40. Communication Protocol	41. Ring	42. Bus
43. Sub-Network	44. Ring	45. WAN
46. International Standard organization	47. Software	48. Hardware
49. Hardware	50. International Business Machine	51. Repeater
52. Bridge	53. Gateway	54. Router
55. Un-Shielded Twisted Pair	56. IEEE 802.3, IEEE 802.4	57. Millions of bits per second
58. LAN	59. Bandwidth	60. Attenuation
61. Local Area Network	62. Spine	63. Internetwork

True/False

1. The components of data communications are source, sender and receiver.
2. A receiver is also called source.
3. A sender is also called sink.
4. A medium that carries the message.
5. A collection of raw facts is called information.
6. Breaking a long message into smaller block is called data routing.
7. All computers are not equally efficient in terms of speed.
8. Transfer of data in the form of digital signals is called digital data transmission.
9. Sound waves are analog signals.
10. There are four modes of data transmission.
11. The full duplex mode is faster than half duplex mode for two-way data transmission.
12. In simplex mode, data communication channel used in both directions.
13. Telephone line is an example of half duplex mode.
14. Information stored on computer can consist of text, number , sound or pictures.
15. Analog signal is measured in hertz.
16. A digital signal is a sequence of voltage represented in binary form.
17. In synchronous transmission data is transmitted character by character.
18. Synchronous transmission is well suited for remote communication between a computer and related devices like card reader and printers.
19. Wire pair is commonly used in local telephone communication.
20. Modem can transfer voice and data simultaneously.
21. The modem converts the digital signal into tones for transmitting through phone lines.
22. Synchronous transmission is much faster than asynchronous.
23. Transmission of signals across a communication medium is called signaling.
24. WAN stands for Wide Area Network
25. WAN is not as fast as LAN
26. WAN is expensive than LAN
27. Telephony software and hardware allow computers to serve as speakerphones, answering machines, and complete voice mail systems.
28. A user can do with voice mail just about anything that can be done with electronic mail, including edit messages electronically.
29. Funds cannot be transferred electronically without an ATM.
30. Bandwidth generally refers to the quantity of information that can be transmitted through a communication medium in a given amount of time.
31. Fiber optic cables use sound waves to carry information at blinding speeds.
32. One way to increase bandwidth in a cable is to increase number of parallel wires in that cable.
33. Bus topology supports a small number of computers
34. All computers or network nodes are connected to a common communication medium in star topology.
35. It is possible to connect computers in different way.
36. In a network maximum five computers can be connected

37. In bus topology communication medium is hub.
38. It is easy to extend bus topology.
39. Each computer is connected to the next computer with the last one connected to the first in star topology.
40. Ring topology is less expensive than star topology.
41. ISO is an example of software protocol.
42. A short message that travels around the communication medium is called token.
43. A network protocol that determines how data is transferred from one place to another is called Hardware network protocol
44. Ethernet is an example of software network protocol
45. Repeater is a device that is used to connect networks of same types to boost the signal
46. A router is a device that connects multiple networks using similar or different protocols
47. Gateway is a device that connects two or more networks with different types of protocols
48. Bridge is used where different types of network are to be joined
49. A collection or two or more networks is called network of networks
50. X23 is the standard used by the most telephone companies of the world.
51. Coaxial cable is the most common cable used in Ethernet (802.3) local area networks.

Answers

1. T	2. F	3. F	4. T	5. F	6. F
7. T	8. T	9. T	10. F	11. T	12. F
13. F	14. T	15. F	16. T	17. F	18. T
19. T	20. T	21. T	22. T	23. T	24. T
25. T	26. T	27. T	28. F	29. F	30. T
31. F	32. T	33. T	34. F	35. T	36. F
37. F	38. T	39. F	40. T	41. F	42. T
43. F	44. F	45. T	46. T	47. T	48. F
49. T	50. F	51. F			

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

14

WINDOWS 2000 PROFESSIONAL: INSTALLATION & CONFIGURATION

Chapter Overview

14.1 Windows 2000 Professional
    14.1.1 Features of Windows 2000 Professional
14.2 Workgroup and Domain
    14.2.1 Workgroup
    14.2.2 Domain
14.3 Disk Partition
    14.3.1 Types of Partitions
    14.3.2 Volumes
14.4 File Systems
    14.4.1 FAT
    14.4.2 FAT32
    14.4.3 NTFS
    14.4.4 CDFS
    14.4.5 Difference between FAT and NTFS
    14.4.6 Disk Management Utility
14.5 Windows Operating System
    14.5.1 Peer-to-peer Workgroup
    14.5.2 Server Centric Network
    14.5.3 Installing Windows 2000 from CD-ROM
    14.5.4 Installing Windows 2000 over a Network
14.6 Windows 2000 Desktop
    14.6.1 Elements of Desktop
    14.6.2 Desktop Properties
    14.6.3 Wallpaper
    14.6.4 Screen Saver
        14.6.4.1 Energy Saver Feature
    14.6.5 Appearance
    14.6.6 Effects
    14.6.7 Display Properties
    14.6.8 Date and Time
    14.6.9 Mouse Properties
    14.6.10 Keyboard Properties
14.7 Shortcut

- 14.7.1 Creating Shortcuts

14.8 Accessibility Features
  - 14.8.1 Accessibility Wizard

14.9 Control Panel
  - 14.9.1 Elements of Control Panel

14.10 Computer Management
  - 14.10.1 Using Microsoft Management Console
  - 14.10.2 Using Administrative Tools

14.11 Using Printers with Windows
  - 14.11.1 Installation of Local Printer
  - 14.11.2 Connecting to a Network Printer
  - 14.11.3 Managing Printer Queue

14.12 Adding and Removing Hardware
  - 14.12.1 Adding Null Modem
  - 14.12.2 Installing a Network Adapter
  - 14.12.3 Configuring Network Adaptor

14.1 Windows 2000 Professional

Windows 2000 Professional is developed by **Microsoft**. It is a reliable and faster operating system. It is used to run software applications, connect to Internet or intranet, access files, printers and network resources.

Windows 2000 Professional is built on Windows NT technology. It has easy-to-use interface. It provides increased flexibility to business users. The Web capabilities allows the user to connect to Internet from anywhere at anytime.

14.1.1 Features of Windows 2000 Professional

Some important features of Windows 2000 Professional are as follows:

1. Easier to Use

Windows 2000 Professional provides an easier way to work with files and folders. It saves all files in My Documents folder on Desktop by default. It also manages a list of recently used files for quick access later.

2. Easier to Manage

Windows 2000 Professional is easier to manage for user and network administrator. The user can easily add and remove programs. He can also sort them by how frequently they are used. Windows 2000 can automatically install and repair some programs.

3. Multitasking

Multitasking is the capability of loading multiple programs in memory and executing them at the same time. Windows 2000 is a multitasking system. A single user can run multiple programs simultaneously. The user can edit a document while listening music.

4. Multi-User

Windows 2000 is a multi-user operating system. It means that many users can use the computer at the same time. These users can be connected to a computer over the Internet.

5. Multiprocessing

Windows 2000 supports multiple processing. Multi-processing is the capability to support and utilize multiple processors in a computer. Two or more processors can execute different programs at the same time. It increases the output of the computer.

6. Plug and Play

Plug-and-play is the capability to detect and configure a device and install appropriate device driver. Windows 2000 has the feature of plug and play. It contains a large number of device drivers. When a new hardware device is attached to computer, it detects the device and installs its driver if it is available.

7. Networking

Windows 2000 provides networking features. It provides the facility to establish, maintain and troubleshoot a network.

8. Latest Hardware Support

Windows 2000 includes the latest drivers to support the latest advancements in computer hardware technology. Windows 2000 now includes hardware support for:

- USB Ports
- Web TV
- Laptop Computers

9. More Reliable

Windows 2000 Professional keeps track of files and components needed by programs. If a file is missing or damaged, Windows will search and reinstall the files automatically.

10. Backup and Recovery

Backup and recovery feature is used to backup data and recover it in case of hard disk failure. Windows 2000 allows back up of files on hard disk or other media

11. More Secure

Windows 2000 Professional can restrict access to your programs, files, and folders, and other resources on the network. It can also filter content.

12. Performance

Windows 2000 Professional is 32 percent faster than Windows 95. It is 27 percent faster than Windows 98. It can execute the most popular business applications with 64 MB of RAM. It is also significantly faster than Windows NT 4.0.

13. Internet Connection Sharing

Windows 2000 Professional allows the user to connect home network or small office network to Internet. A connection can be established using dial-up or broadband connection.

14. Windows Update via Internet

The user can update Windows operating system via Internet. The **Windows Update Wizard** can be used to download and install the latest updates for Windows and software.

15. Fewer Bugs

Microsoft has fixed over 1,800 bugs in Windows 2000. That makes Windows 2000 more reliable and less prone to crashes than Windows 98.

14.2 Workgroup and Domain

There are two prevalent modes of grouping networked computers in Window NT and later versions of operating systems:

14.2.1 Workgroup

In a **workgroup model**, the network computers located in one place are physically connected to each other. It is used for a group of people who are working on a task. In this environment, the computer of all persons in the group are connected to the network. It allows them to send email to one another, share data files and schedule setting.

A workgroup is a set of one or more computers with a workgroup name. Any user can join a workgroup by specifying the workgroup name. No special permission is required to join a workgroup. There is a local account database on each client of network in workgroup. It is used to maintain user accounts and security settings locally. There is no centralized account database or centralized security.

Advantages

Some advantages of workgroup are as follows:

1. It is simple to design
2. It is easy to share resources
3. It is convenient for a small number of users.

Disadvantages

Some disadvantages of workgroup are as follows:

1. No centralized management
2. Duplication of accounts
3. Inefficient for a large network

14.2.2 Domain

A domain is a larger collection of computers defined by a network administration. It is a logical group of computers and devices on a network that are administered as a unit with common rules and procedures. The domains are identified by IP address within the Internet.

A domain model follows a hierarchical system. Each domain has its own control but is also a part of a large domain. For example, all computers in Sales department are part of one domain. The sales domain is a subset of Marketing domain. The other domains in marketing domain may be advertising and marketing domain is part of overall organization domain.

Advantages

Some advantages of domain are as follows:

1. Centralized management
2. Security management is central. User access to computers can be controlled.

Disadvantages

Some disadvantages of domain are as follows:

1. Administration is more complex
2. Sharing resources is complex

14.3 Disk Partition

The disk partition is a disk management technique. It is a logical division of hard disk. A hard disk can contain one or more partitions. Partitioning is useful for the user to use multiple operating systems.

14.3.1 Types of Partitions

Windows usually creates the following two basic types of partitions:

- Primary Partition
- Extended Partition

1. Primary Partition

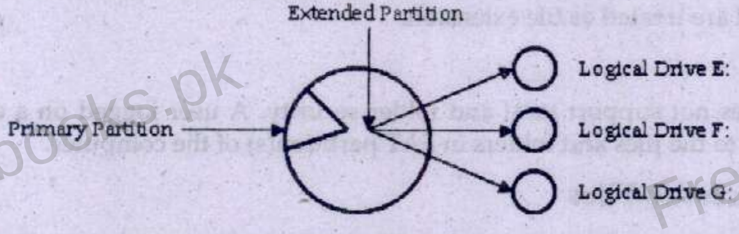
The **primary partition** is a type of partition that can be used as **system partition**. The system partition contains hardware-specific files required to load Windows 2000.

A primary partition can be created to occupy the entire hard disk or a potion. Any hard disk can have a maximum of four partitions. All four partitions can be primary partitions. Each primary partition must be formatted after partitioning. The primary partition should be formatted as a single logical drive only. It cannot be subdivided into multiple logical drives.

2. Extended Partition

The **extended partition** is a type of partition that can be divided into subpartitions. The sub partitions are known as **logical partitions**. It is done to use multiple operating systems. A hard disk can have only one extended partition.

The extended partitions can also be formatted to create drives. Any number of drives can be created on an extended partition unlike primary partitions. Each drive can be designated by an alphabet.



The above figure shows a hard disk with one primary partition and one extended partition. The extended partition is subdivided into logical drives. One logical drive can be formatted with **FAT** and the other with **NTFS** from the same extended partition.

14.3.2 Volumes

A fixed amount of storage space on a medium such as hard disk is known as **volume**. Each volume has a **label**. A volume may consists of more than one disk also. A volume contains **root directory** that has a table of disk contents. It also has the **allocation tables**. It also contains entries about the occupied spaces.

14.4 File Systems

Everything is stored as files in a computer system. The files can be **data files** or **application files**. Each operating system has its own way of organizing data internally. The operating system performs this management with the help of a program called **file system**. The type of file system is used to determine how data and programs are accessed. It also determines the level of accessibility available to users.

Some important operating systems and their supported file systems are as follows:

Operating System	File System Supported
Windows 2000	FAT,NTFC,CDFS
Windows 95	FAT, CDFS
Windows 3.x and 3.1 x	FAT,CDFS
MS-DOS	FAT,CDFS

14.4.1 FAT

FAT stands for **file allocation table (FAT)**. The file system has been used since the advent of PC. The FAT file system used by Windows 2000 is a modified version of FAT file system used by MS-DOS.

Features of FAT

Some important features of FAT are as follows:

1. Naming Conventions

The naming conventions followed by FAT are as follows:

- FAT file system used by MS-DOS provides file name of only 8 characters long.
- FAT file system used by Windows 2000 supports long file name. The full path of file including filename can be up to 255 characters long.
- File names can contain any character except " / \ [] = , = ^ ?a ""
- File names should begin with alphanumeric characters.
- File names can contain spaces and multiple periods. The characters after the last period are treated as file extension.

2. Security

FAT does not support local and folder security. A user logged on a computer locally has full access to the files and folders in FAT partition(s) of the computer.

3. Quick Access to Files

FAT provides quick access to files. The speed of file access depends on file type, file size, partition size, fragmentation and number of files in a folder.

- File Type: The text files takes less accessed time compared to graphics file.
- File Size: The access speed is better if the file is of smaller size.
- Partition Size: The smaller partition size gives greater access speed. The size smaller than 500MB have better file access in FAT. NTFS is required for bigger partition sizes.
- Fragmentation: The small portions of free spaces on a disk are called fragments. The access time is better if there is less fragmentation.
- Number of files in a folder: The access speed increases if there is less number of files in the folder.

14.4.2 FAT32

FAT32 is an advanced version of FAT file system. It can be used on drives from 512 MB to 2TB in size. One of the most important feature of FAT and FAT32 is that they offer compatibility with operating systems other than Window 2000 also.

Features of FAT32

FAT 32 has the following features:

1. Partition Size

FAT32 increases number of bits used to address cluster. A cluster is a set of sectors. It reduces size of each cluster. It supports larger disk (upto 2TB) and better storage efficiency.

2. Access Speed

FAT32 provides good file access in partition sizes less than 500 MB or greater than 2GB. It provides better disk space utilization.

14.4.3 NTFS

Windows 2000 Professional fully supports NTFS. It has the following characteristics:

1. Naming Conventions

The naming conventions followed by NTFS are as follows:

- File names can be up to 255 characters
- File names can contain most characters except " / \ < > * | :
- File names are not case sensitive

2. Security

NTFS provides file and folder security. Files and folders are safer than FAT. Security is maintained by assigning **NTFS** permissions to files and folders. Security is maintained at the local level and the network level. The permissions can be assigned to individual files and folders. Each file or folder in an NTFS partition has an **Access Control List**. It contains the users and group **security identifier (SID)** and the privileges granted to them.

3. Partition Size

The NTFS partition and file sizes are much bigger than FAT partitions and files. The maximum size of an NTFS partition or file can be 16 exabytes. However, the practical limitation is two terabytes. The file size can be in the range of 4GB to 64 GB.

4. File Compression

NTFS provides file compression of as much as 50%.

5. High Reliability

NTFS is highly reliable. It is a recoverable file system. It uses transaction logs to update the file and folders logs automatically. The system also has a great amount of **fault tolerance**. It means that if a transaction fails due to power or system failure, the logged transactions are used to recover the data.

6. Bad Cluster Mapping

NTFS supports bad-cluster mapping. It means that file system detects bad clusters or areas of disk with errors. If there is any data in those clusters, it is retrieved and stored on another area. The bad clusters are marked to prevent data storage in those areas in future.

14.4.4 CDFS

CDFS is the file system for compact disk storage media. Windows 2000 also supports it. It is a 32-bit program helping in interpreting data.

14.4.5 Difference between FAT and NTFS

FAT	NTFS
It provides no security if user logs in locally. The file and folder level security permission does not exist.	It provides security for both local and remote users. The security is provided to the level of files and folders.
It supports only 8 characters long filename.	It supports 255 characters long file name.
It is suitable for partition that is less than 500MB	It is suitable for partition that is greater than 500MB
Partitions and file size can be 4 GB maximum.	Partitions size can be a maximum of 16 exabytes.
It supports no file compression.	It supports file compression.
Disk can get fragmented thus slowing down the access.	It provides lesser possibility of fragmentation.
It is not very reliable since it does not support bad cluster mapping	It is highly reliable since it supports bad cluster mapping and transaction logging.

14.4.6 Disk Management Utility

Disk Management utility provides a graphical interface to view and maintain all hard drives and CD drives. The user can find the size of the drive. Disk management also indicates the drive that contains the system partition.

14.5 Windows Operating System

Windows 2000 is the latest operating system. Windows operating systems are available in two categories called Professional and Server.

14.5.1 Peer-to-peer Workgroup

In a peer-to-peer network, computers are connected to one another via cables. The users can share files and devices such as printers. Peer-to-peer networks are commonly used in a small organization with fewer than five computers.

14.5.2 Server Centric Network

In a server centric network environment, **client computers are connected to a server. Server is more powerful than standard computers. It stores and delivers information, devices and software programs to clients. Server centric networks are typically used in large organizations.**

Hardware Requirements

The hardware requirements for installing Windows 2000 Professional are as follows:

Components	Requirements
CPU	Pentium based
Memory	32 MB
Hard disk	One or more hard disks of minimum 650 MB (2GB recommended) on the partition that will contain the system files
Networking	Network adapter card

Display	Video display adapter and monitor with Video Graphics Adapter resolution or higher
Other Devices	CD-ROM Drive, 12X or faster recommended (not required for installing windows 2000 over a network)
Accessories	Keyboard and mouse or other pointing device

14.5.3 Installing Windows 2000 from CD-ROM

The installation of Windows 2000 Professional from CD-ROM consists of four stages:

1. Run the Setup Program

The setup program prepares hard disk for later stages of installation. It copies necessary files to run the setup wizard. This stage of setup runs in text mode.

2. Run the Setup wizard

The setup wizard requests setup information about the computer. It includes names, passwords, regional settings etc. This stage of setup runs in graphics mode.

3. Install Windows Networking

The setup wizard requests for networking information. It then installs the networking components so that the computer can communicate with other computers on the network.

4. Complete the setup program

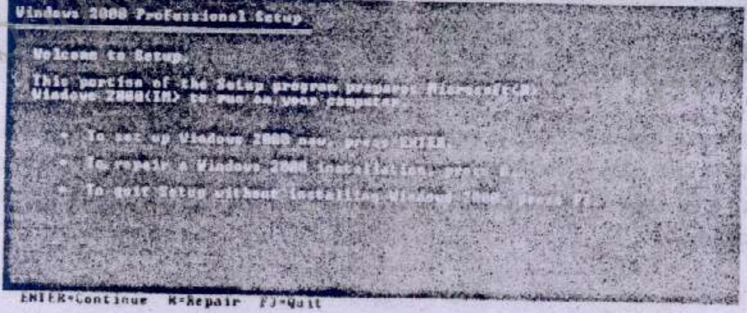
The setup copies files to the hard disk, registers components and configures computer. The system restarts after installation completes.

Installing Windows 2000 Professional

The following procedure is used to install Windows 2000 Professional:

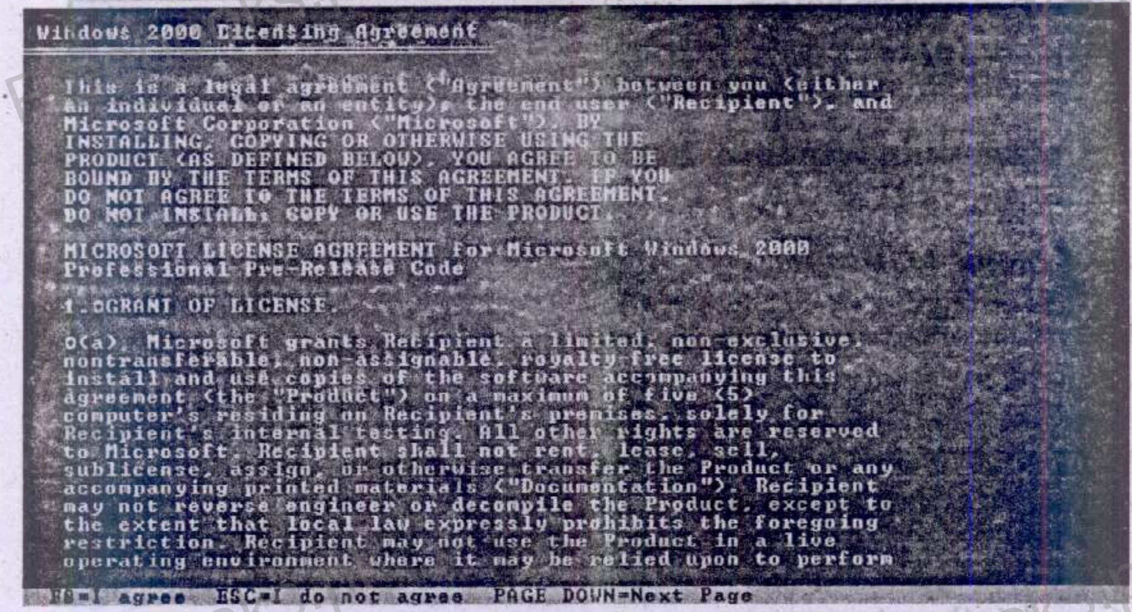
Running the Setup Program

1. Boot the computer and insert Windows 2000 CD in CD-ROM drive.
2. Access the CD-ROM drive from **DOS** prompt.
3. Type "CD I386" and press **Enter** from prompt to change directories to I386.
4. Type **WINNT** from \I386> prompt and press **Enter**. The Windows 2000 Professional setup dialog box will appear.
5. Press **Enter** to accept the default path for Windows 2000 distribution files. It will take a few minutes to copy the files.
6. Remove floppy disks from computer and press Enter to restart the computer. The computer will restart and **Welcome to Setup** screen will appear.
7. Press Enter to setup Windows 2000. The licensee agreement dialog appears.



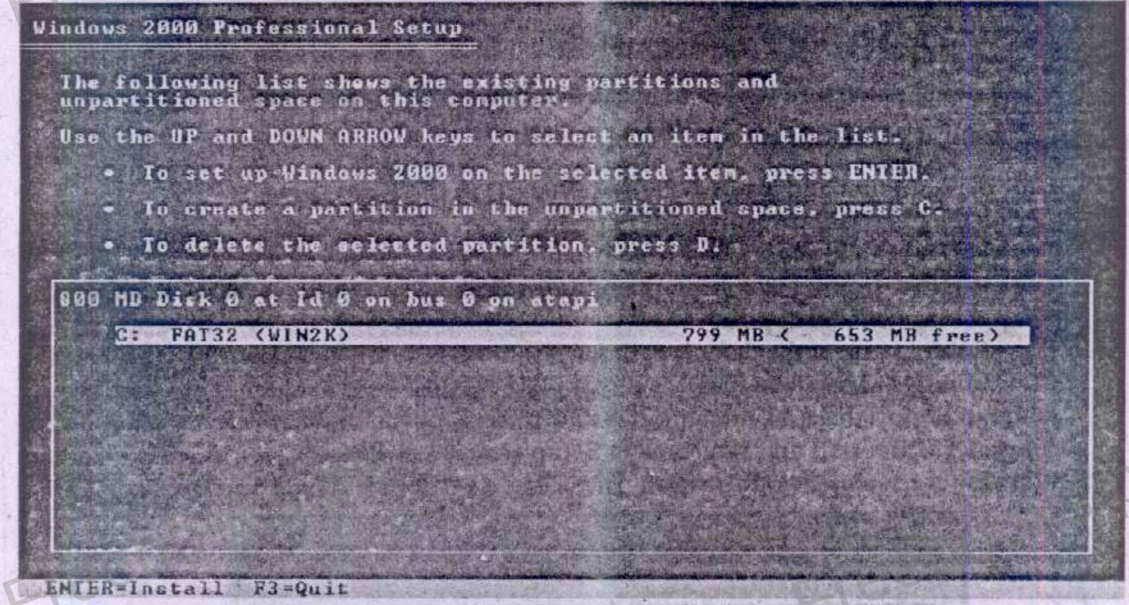
```
Windows 2000 Professional Setup

  Welcome to Setup.

  This portion of the Setup program prepares Microsoft(R)
  Windows 2000(TM) to run on your computer.

     •  To set up Windows 2000 now, press ENTER.

     •  To repair a Windows 2000 installation, press R.

     •  To quit Setup without installing Windows 2000, press F3.

 ENTER=Continue  R=Repair  F3=Quit
```

8. Scroll down to the bottom of page and press **F8** to agree license terms to continue.



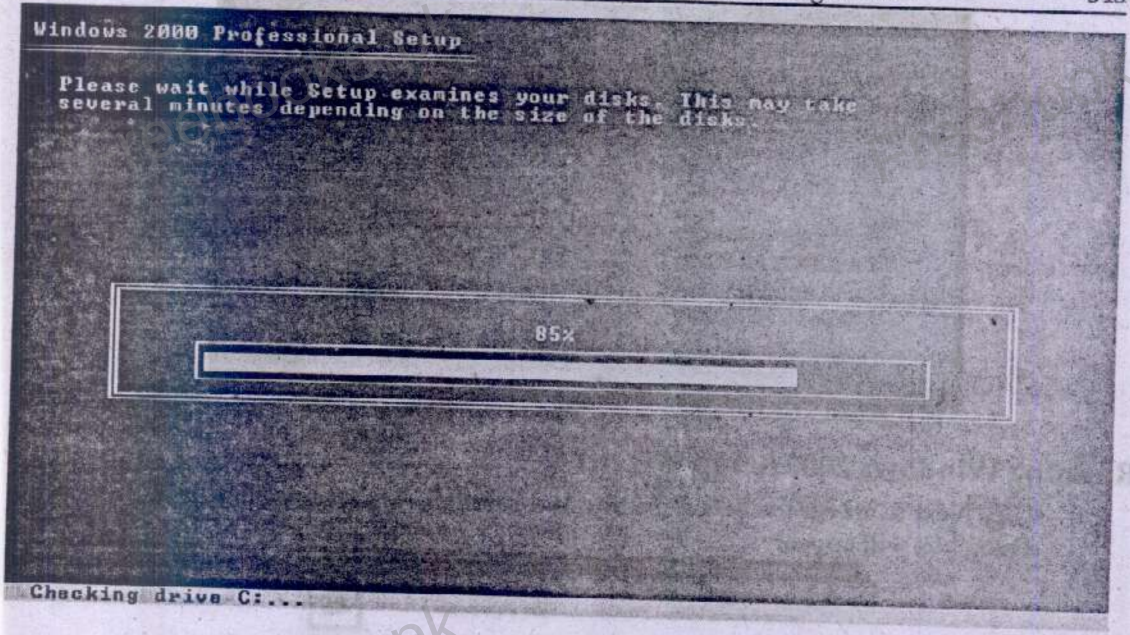
```
Windows 2000 Licensing Agreement

  This is a legal agreement ("Agreement") between you (either
  an individual or an entity), the end user ("Recipient"), and
  Microsoft Corporation ("Microsoft"). BY
  INSTALLING, COPYING OR OTHERWISE USING THE
  PRODUCT (AS DEFINED BELOW), YOU AGREE TO BE
  BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU
  DO NOT AGREE TO THE TERMS OF THIS AGREEMENT,
  DO NOT INSTALL, COPY OR USE THE PRODUCT.

  MICROSOFT LICENSE AGREEMENT for Microsoft Windows 2000
  Professional Pre-Release Code

  1.0GRANT OF LICENSE.

  0(a). Microsoft grants Recipient a limited, non-exclusive,
  nontransferable, non-assignable, royalty-free license to
  install and use copies of the software accompanying this
  agreement (the "Product") on a maximum of five (5)
  computer's residing on Recipient's premises, solely for
  Recipient's internal testing. All other rights are reserved
  to Microsoft. Recipient shall not rent, lease, sell,
  sublicense, assign, or otherwise transfer the Product or any
  accompanying printed materials ("Documentation"). Recipient
  may not reverse engineer or decompile the Product, except to
  the extent that local law expressly prohibits the foregoing
  restriction. Recipient may not use the Product in a live
  operating environment where it may be relied upon to perform
F8=I agree  ESC=I do not agree  PAGE DOWN=Next Page
```

9. Specify **C:** partition to setup Windows 2000 and press **Enter**.

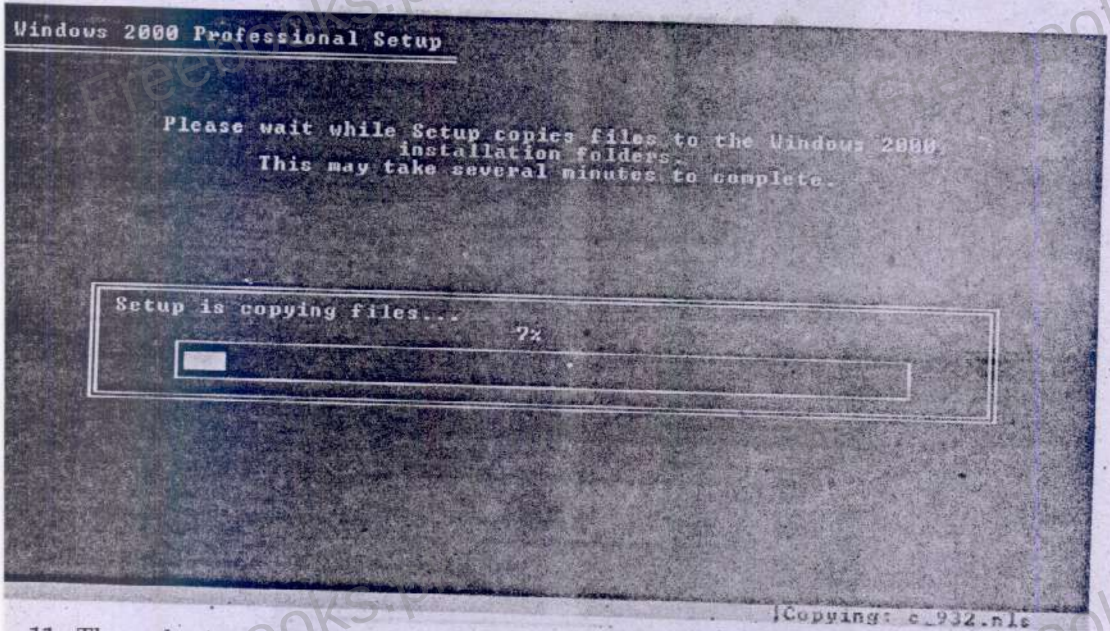


```
Windows 2000 Professional Setup

  The following list shows the existing partitions and
  unpartitioned space on this computer.

  Use the UP and DOWN ARROW keys to select an item in the list.

     •  To set up Windows 2000 on the selected item, press ENTER.

     •  To create a partition in the unpartitioned space, press C.

     •  To delete the selected partition, press D.

  800 MB Disk 0 at Id 0 on bus 0 on atapi

       C:   FAT32 (WIN2K)                          799 MB (   653 MB free)

ENTER=Install  F3=Quit
```

10. Choose to leave the current file system intact and press Enter to continue. The setup will examine the disk. The Windows installation files will be copied to installation folder that will take few minutes. The computer will automatically reboot after the files are copied.

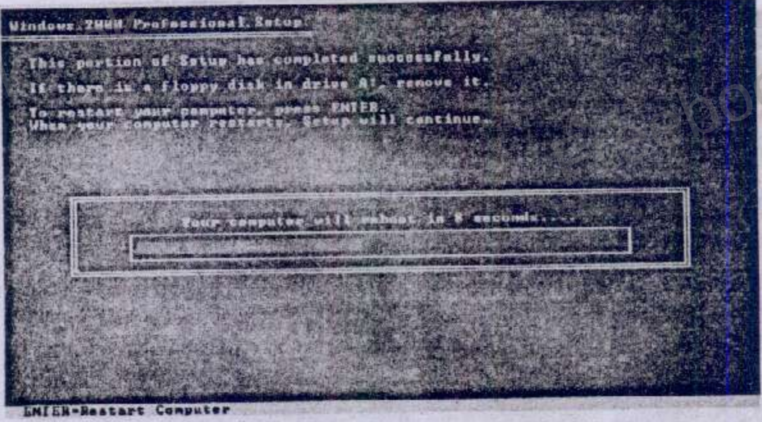


```
Windows 2000 Professional Setup

 Please wait while Setup examines your disks. This may take
 several minutes depending on the size of the disks.



                              85%

Checking drive C:...
```



```
Windows 2000 Professional Setup

        Please wait while Setup copies files to the Windows 2000
                       installation folders.
             This may take several minutes to complete.


  Setup is copying files...
                            7%

                                            Copying: c_932.nls
```

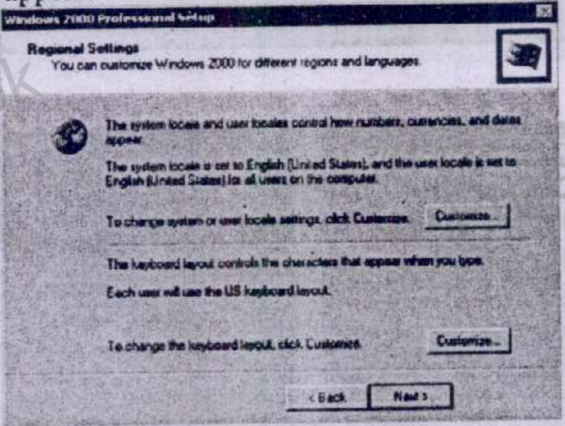
11. The welcome to the setup wizard will appear after the computer reboots.
12. Click Next to continue. The installation will continue automatically if no option is selected in 10 seconds.
13. The setup will automatically detect and install drivers on computer. The process will take a few minutes. The regional setting dialog box appears.



```
Windows 2000 Professional Setup

This portion of Setup has completed successfully.

If there is a floppy disk in drive A:, remove it.

To restart your computer, press ENTER.
When your computer restarts, Setup will continue.

          Your computer will reboot in 8 seconds....

ENTER=Restart Computer
```

Running Windows 2000 Setup Wizard

1. Click **Next** to accept the default setting and continue. The **Personalize Your Software** dialog box will appear.



Windows 2000 Professional Setup

Regional Settings
You can customize Windows 2000 for different regions and languages.

The system locale and user locales control how numbers, currencies, and dates appear.

The system locale is set to English (United States), and the user locale is set to English (United States) for all users on the computer.

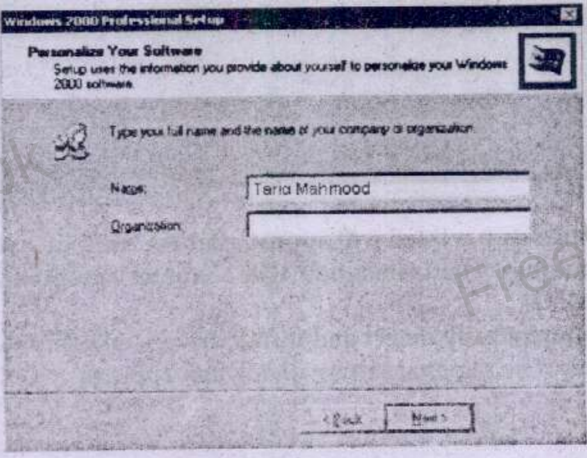
To change system or user locale settings, click Customize.

The keyboard layout controls the characters that appear when you type.

Each user will use the US keyboard layout.

To change the keyboard layout, click Customize.

2. Type your name and organization.
3. Click **Next** button.



Windows 2000 Professional Setup

Personalize Your Software
Setup uses the information you provide about yourself to personalize your Windows 2000 software.

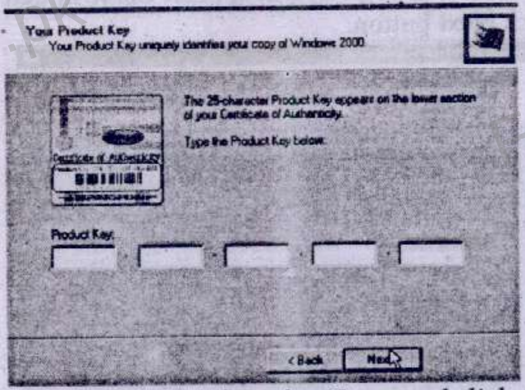
Type your full name and the name of your company or organization.

Name: Tariq Mahmood

Organization:

4. The Setup displays product key page.
5. Input the 25-character Product Key that can be found in Windows 2000 packaging.

6. Click **Next** to continue.



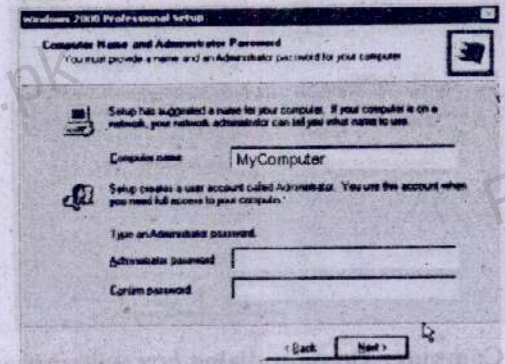
Your Product Key
Your Product Key uniquely identifies your copy of Windows 2000

The 25-character Product Key appears on the lower section of your Certificate of Authenticity.

Type the Product Key below:

Product Key:

7. The **Computer Name and Administration Password** dialog box appears. Type the computer name.
8. Specify an administrator password is necessary.
9. Click **Next** button.



Windows 2000 Professional Setup

Computer Name and Administrator Password
You must provide a name and an Administrator password for your computer

Setup has suggested a name for your computer. If your computer is on a network, your network administrator can tell you what name to use.

Computer name: MyComputer

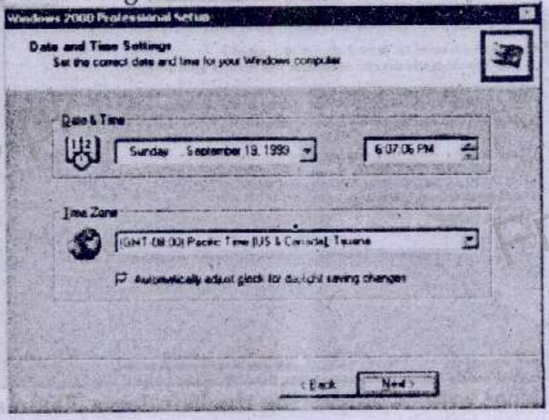
Setup creates a user account called Administrator. You use this account when you need full access to your computer.

Type an Administrator password.

Administrator password:

Confirm password:

10. The modem dialing information dialog box will appear if plug and play modem is attached.
11. Specify the setting for the modem and click **Next** button. The **Date and Time Setting** dialog box appears.
12. Verify that all of the settings are correct and click **Next** button.



Windows 2000 Professional Setup

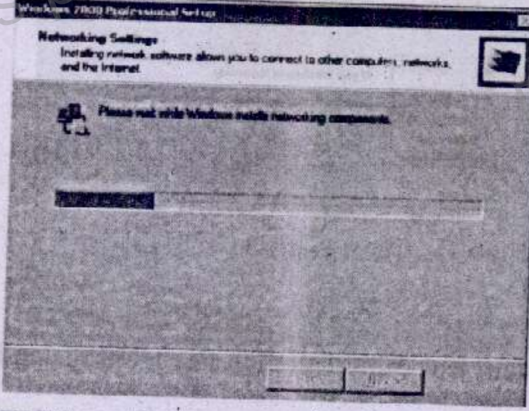
Date and Time Settings
Set the correct date and time for your Windows computer

Date & Time
Sunday , September 19, 1999
6:07:06 PM

Time Zone
(GMT-08:00) Pacific Time (US & Canada); Tijuana

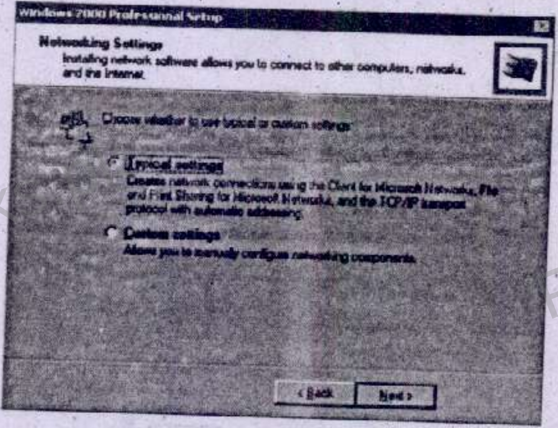
☑ Automatically adjust clock for daylight saving changes

13. The **Networking Setting** dialog box will appear. Confirm that **Typical Setting** button is selected and click **Next** button.


Windows 2000 Professional Setup
Networking Settings

Installing network software allows you to connect to other computers, networks, and the Internet.

Please wait while Windows installs networking components.


Windows 2000 Professional Setup
Networking Settings
< Back
Next >

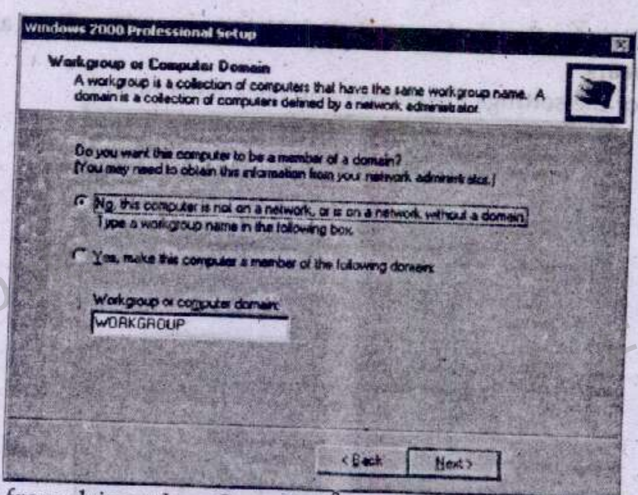
Installing network software allows you to connect to other computers, networks, and the Internet.

Choose whether to use typical or custom settings:

● Typical settings
Creates network connections using the Client for Microsoft Networks, File and Print Sharing for Microsoft Networks, and the TCP/IP transport protocol with automatic addressing.

○ Custom settings
Allows you to manually configure networking components.

14. The **Workgroup or Computer Domain** dialog box will appear. Select the first radio button and leave the default workgroup name.
15. Click **Next** button. The setup components will be installed that takes several minutes.


Windows 2000 Professional Setup
Workgroup or Computer Domain
< Back
Next >

A workgroup is a collection of computers that have the same workgroup name. A domain is a collection of computers defined by a network administrator.

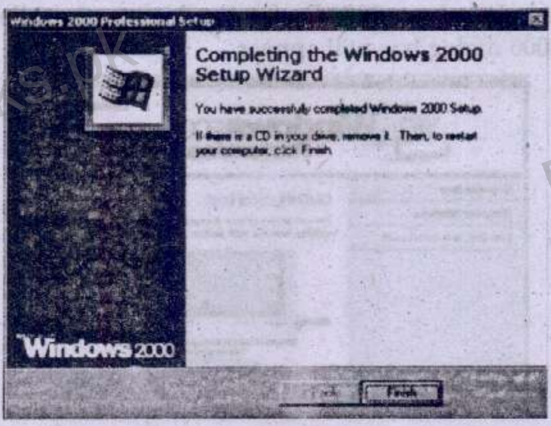
Do you want this computer to be a member of a domain?
(You may need to obtain this information from your network administrator.)

● No, this computer is not on a network, or is on a network without a domain.
Type a workgroup name in the following box.

○ Yes, make this computer a member of the following domain:

Workgroup or computer domain: WORKGROUP

16. Remove CD from drive when **Completing the Windows 2000 Setup Wizard** appears and click **Finish** button. The computer will restart.



Windows 2000 Professional Setup
Windows 2000
Finish

Completing the Windows 2000 Setup Wizard

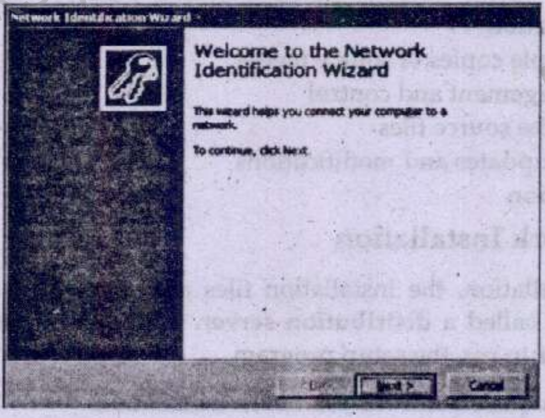
You have successfully completed Windows 2000 Setup.

If there is a CD in your drive, remove it. Then, to restart your computer, click Finish.

17. When computer reboots, choose Microsoft Window 2000 from boot selection option by pressing Enter.

Running Network Identification Wizard

1. The Windows 2000 Professional starts and displays **Welcome to the Network Identification Wizard** dialog box. Click **Next** to continue. The **Users of This Computer** dialog box will appear.



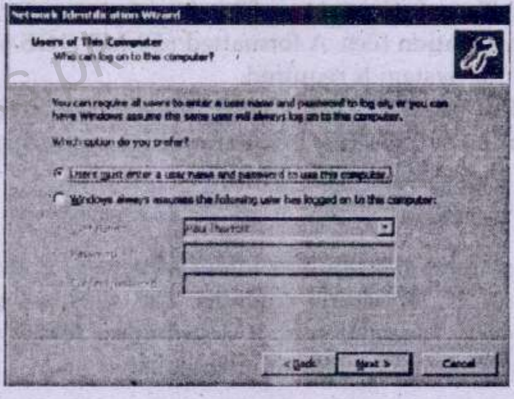
Network Identification Wizard
< Back
Next >
Cancel

Welcome to the Network Identification Wizard

This wizard helps you connect your computer to a network.

To continue, click Next.

2. Click **Users must enter a user name and password to use this computer** option.
3. Click **Next** to continue.



Network Identification Wizard
< Back
Next >
Cancel

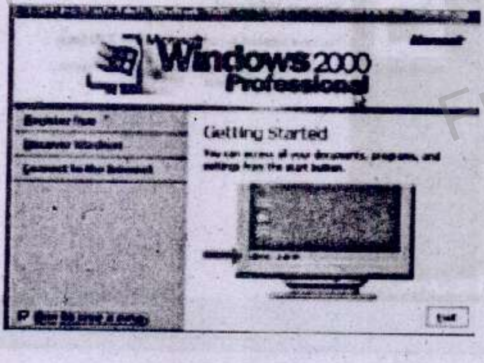
Users of This Computer
Who can log on to this computer?

You can require all users to enter a user name and password to log on, or you can have Windows assume the same user will always log on to this computer.

Which option do you prefer?

● Users must enter a user name and password to use this computer.
○ Windows always assumes the following user has logged on to this computer:

4. Click **Finish** button. Windows 2000 Professional is now installed and **Getting started with Windows 2000** dialog box will appear.



14.5.4 Installing Windows 2000 over a Network

Windows 2000 installation is a lengthy and time-consuming process. The user can install Windows 2000 of many computers more easily over a network instead of installing from CD ROM individually.

Benefits of Networks Installation

Windows installation across the network provides many benefits:

- Easy mass installation
- No need of multiple copies of source media
- Centralized management and control
- Secure access to the source files
- Easier access for updates and modifications
- Speed of installation

Preparing for Network Installation

In a network installation, the installation files are stored in a shared location on a network file server. It is called a **distribution server**. The user can connect to distribution server from any computer to run the setup program.

Following are the requirements for a network installation:

- **Distribution Server:** Distribution server contains installation files from i386 folder on Windows 2000 Professional CD-ROM. The folder should be shared across network.
- **FAT partition on Target Computer:** The target computer requires a formatted partition to copy installation files. A formatted 650-MB (1GB or larger recommended) partition with FAT file system is required.
- **Network Client:** The client or clients need to be started from network software to connect to distribution server. The installation files are accessed from shared folder after connecting to the distribution server.

Creating a Distribution Server

A distribution server stores distribution folder structure. It requires about 300MB of space. The distribution folder structure contains the files to install Windows 2000 Professional over the network. Multiple sets of distribution folder can also be created to install Windows 2000 on multiple computers simultaneously.

The following procedure is used to create distribution server:

1. Log on to the server as Administrator OR connect to the server to create distribution file structure.
2. Create a folder on the server named W2000p.
3. Copy the contents of i386 folder from Windows 2000 Professional CD-ROM to the newly created folder.
4. Create an **OEM** subfolder in **W22000p** folder. It is used to store applications, drivers or utility to be copied to the target computer.

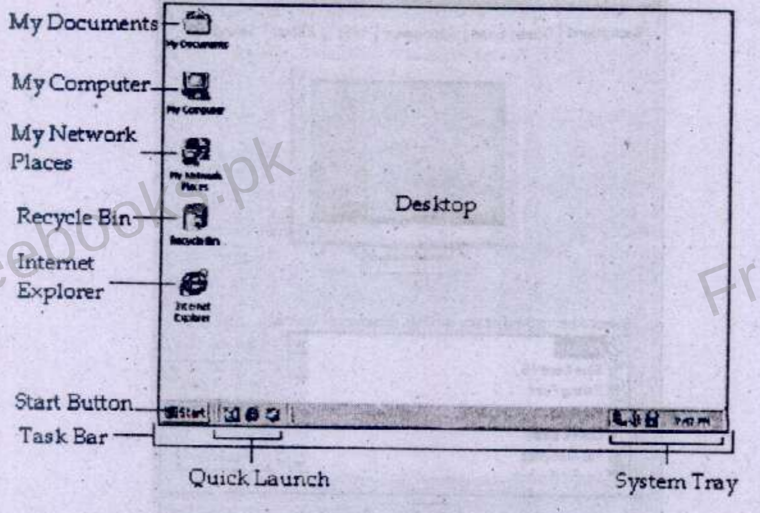
Performing Installation over Network

The following procedure is used to install Windows 2000 over the network.

1. Boot from the network client on the target computer.
2. Connect to distribution server after starting the network client on target computer.
3. Connect to the shared folder on distribution server that contains the installation files of Windows 2000 Professional.
4. Run **winnt.exe** or **Winnt32.exe** to start setup. The **winnt.exe** is used for installation using windows 3.x on source system. The **winnet32.exe** is used for an installation using Windows 95, 98, NT4 or 2000. Both files reside in shared folder on distribution server. The following steps are performed when one of the files is executed:
   - A temporary folder Win_nt.~1s is created on target computer.
   - The Window 2000 installation files are copied from shared folder on the distribution server to Win_nt~1s folder on target computer.
5. The setup restarts target computer and begins installing Windows 2000 Professional. The remaining steps are same as installation using CD-ROM.

14.6 Windows 2000 Desktop

The on-screen work area that contains windows, icons, menus and dialog boxes is called **desktop**. The desktop is the entering point in Windows. Desktop is the first object that appears when Windows is started. Icons are small graphic that represents programs, drives, folders, and documents. It is used to start program, open a document, or access the contents of a drive or folder. You can also change the look of the desktop.



14.6.1 Elements of Desktop

Different elements of Desktop are as follows:

1. Icons

An icon is a small graphic that represents a program, drive, folder or document. Windows operating system uses icons frequently to make the environment user-friendly. Every icon also has a label that can be changed. The user can activate an icon to start a program, open a document or access the contents of a drive or folder.

Some important icons that appear on Desktop are My Computer, My Documents, My Network Places, Recycle Bin and Internet Explorer.

2. Taskbar

The taskbar appears at the bottom of computer screen by default. It contains Start button at the left side. The taskbar may also contain shortcuts to programs or folders. It displays computer clock time and background programs like virus scanners etc. at right side.

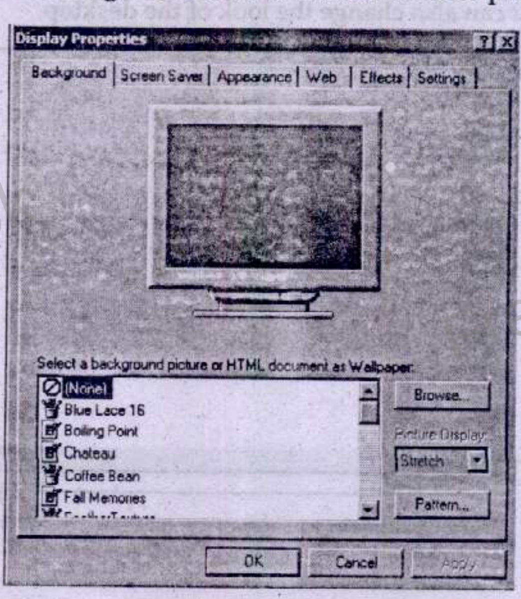
3. Start Button

Start button is used to access most of the programs installed on the computer. It is used to perform the following tasks easily:

- Open or search a document
- Change Windows settings
- Manage files
- Getting help
- Maintain system etc.

14.6.2 Desktop Properties

The Windows operating system provides a variety of options to customize graphical interface. The options include color, font size, display resolution, icons and taskbar options etc. The **Display Properties** dialog box is used to customize these properties.



Display Properties

Select a background picture or HTML document as Wallpaper:

(None)
Blue Lace 16
Boiling Point
Chateau
Coffee Bean
Fall Memories

Picture Display: Stretch

Each user of the computer can customize the desktop properties according to his personal interest. The following procedure is used to customize desktop properties:

1. Right-click an empty area on Desktop. A popup menu will appear.
2. Choose **Properties** from the menu. The **Display Properties** dialog box will appear.

14.6.3 Wallpaper

Wallpaper is the background that appears on desktop. The background may have single color or pattern. It can also display a picture.

Change Wallpaper

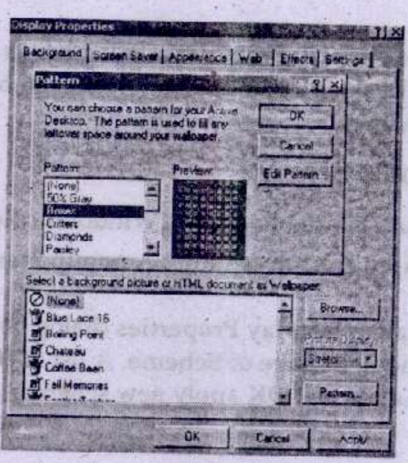
The following procedure is used to change wallpaper:

1. Right-click an empty area on Desktop. A popup menu will appear.
2. Click **Properties**. A **Display Properties** dialog box will appear.
3. Click on **Background tab**.
4. Select any wallpaper from **Select an HTML Document or Picture box.**
5. Click **Apply**.
6. Click **OK**. The selected wallpaper will be applied.

Apply Pattern

The following procedure is used to apply pattern:

1. Right-click an empty area on Desktop. A popup menu will appear.
2. Click **Properties**. A **Display Properties** dialog box will appear.
3. Click on **Background tab.**
4. Click **Pattern**. The **Pattern** dialog box will appear.
5. Select a patter from **Pattern**; list box and click **OK**.
6. Click **Apply** and then OK.



You can choose a pattern for your Active Desktop. The pattern is used to fill any leftover space around your wallpaper.

Display Properties
Background
Screen Saver
Appearance
Web
Effects
Settings
Pattern
OK
Cancel
Edit Pattern...
Pattern:
Preview
(None)
50% Gray
Boxes
Critters
Diamonds
Paisley
Select a background picture or HTML document as Wallpaper
(None)
Blue Lace 16
Boeing Point
Chateau
Coffee Bean
Fall Memories
Browse...
Picture Display:
Stretch
Pattern...
OK
Cancel
Apply

14.6.4 Screen Saver

A **screen saver** is a moving image that appears on computer screen if the computer is not used for specified period of time. The screen saver saves computer screen from possible damage called **burn-in**. The damage may occur if the screen displays a static image for a long time. The screen saver displays a moving image to prevent this damage. The screen saver disappears as the user presses any key or mouse button or moves the mouse.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

Two important advantages of screen saver are:

- It controls power supply to the monitor
- It protects hard disk by turning it off if the user chooses this option.

The following procedure is used to set screen saver:

1. Right-click anywhere on desktop. A popup menu will appear.
2. Click on **Properties**. A **Display Properties** dialog box will appear.
3. Click on **Screen Saver** tab.
4. Select the desired screen saver from the list.
5. Click on **Settings** to set display options associated with screen saver.
6. Checkmark **Password Protected** to set a password for screen saver. It prevents the screen saver from being turned off unless a password is entered.
7. Click on **Change** to set the password.
8. Set the number of minutes to start screen saver.
9. Click on **OK**. The screen saver will be applied.

You can see a preview of the screen saver by clicking on the **Preview button**.

14.6.4.1 Energy Saver Feature

The energy saver feature allows control over the power supplied to the monitor. The following procedure is used to change the energy saving option:

1. Right click on an empty area on desktop. A popup menu will appear.
2. Select **Properties**. The **Display Properties** dialog box will appear.
3. Click on **Power** button. The power management page will appear.
4. Choose predefined options from the list
5. Click **Apply** to select the option **OR** click **OK**.

14.6.5 Appearance

The **Appearance** tab of **Display Properties** dialog box is used to change color, size and fonts of Windows interface. Desktop color and icon fonts can be changed using this option. It provides difference color schemes.

Changing Windows Color Scheme

The following procedure is used to change Windows color scheme:

1. Right click on an empty area on desktop. A popup menu will appear.
2. Choose **Properties**.
3. Select **Appearance** tab in **Display Properties** dialog box.
4. Click the arrow at the right edge of **Scheme**. A list of color schemes will appear.
5. Choose any scheme and click **OK** apply new scheme.

14.6.6 Effects

The **Effects** tab of **Display Properties** dialog box is used to change the icons that represent My Computer, Network Places and empty and full Recycle Bin. The icons can be enlarged or reduced. The number of colors of the icons can also be changed.

Changing the Icon Settings

The following procedure is used to change the icon settings:

1. Right click on an empty area on desktop. A popup menu will appear.

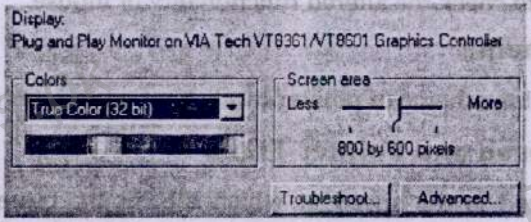
2. Choose **Properties**. The Display Properties dialog box will appear.
3. Select **Effects** tab.
4. Select the desktop icon to be changed.
5. Click on **Change Icon** button. A dialog box will appear containing current icon and horizontal scrolling window with other graphical icons.
6. Select any icon and click **OK** to apply the change.

14.6.7 Display Properties

Display property is used to make changes to display resolution, number of supported colors and monitor type. It appears in **Setting** tab of **Display Properties** dialog box. The resolution and number of colors depend on the type of graphics card and display driver.

The following procedure is used to change display properties:

1. Right click on empty area on desktop. A popup menu will appear.
2. Choose **Properties**. The Display Properties dialog box will appear.
3. Select **Setting** tab.
4. Drag the mouse on the screen area slider and choose the appropriate resolution.



Display:
Plug and Play Monitor on VIA Tech VT8361/VT8601 Graphics Controller

Colors: True Color (32 bit)

Screen area: 800 by 600 pixels

14.6.8 Date and Time

The following procedure is used to change date/ time:

1. Click on **Start** button.
2. Highlight **Settings**. A submenu will appear.
3. Click on **Control Panel**. The Control Panel will open.
4. Click on **Date/Time**. The **Date/Time Properties** dialog box will appear.
5. Select the month and year in **Date** frame.
6. Select current month from **Month** field.
7. Type the year in **Year** field **OR** use arrows next to the field to move forward or backward until to select the current year.
8. **Time** field is divided into four segments: hour, minutes, seconds and AM and PM.
9. Enter values for hour, minutes, seconds and AM/PM.
10. Select the correct time zone from **Time Zone** list.
11. Click **Apply**. The time will be applied.

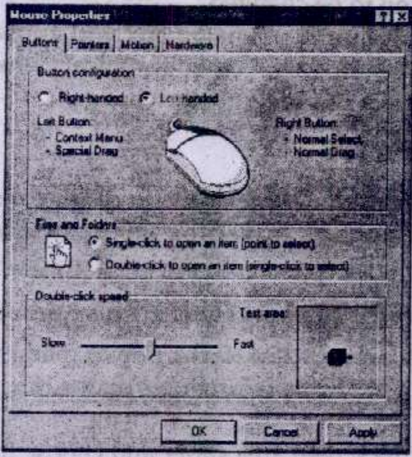
14.6.9 Mouse Properties

Customizing Mouse Properties

The following procedure is used to customize mouse properties:

1. Click **Start** button on the task bar.
2. Point to **Settings > Control Panel**. Control Panel window opens.
3. Double click **Mouse** icon. **Mouse Properties** dialog box opens.

4. Click **Buttons** tab. Click left-handed radio button under **Button configuration** and click to open an item under **File and Folders** heading.


Mouse Properties
Buttons | Pointers | Motion | Hardware
Button configuration
○ Right-handed ● Left-handed
Left Button:
- Context Menu
- Special Drag
Right Button:
- Normal Select,
Normal Drag
Files and Folders
● Single-click to open an item (point to select)
○ Double-click to open an item (single-click to select)
Double-click speed
Slow
Fast
Test area:
OK
Cancel
Apply

5. Press **OK** to check the altered functionality of mouse.

Mouse Pointer Scheme

You can change the scheme of pointer by clicking on **Pointer** and then **Scheme**.

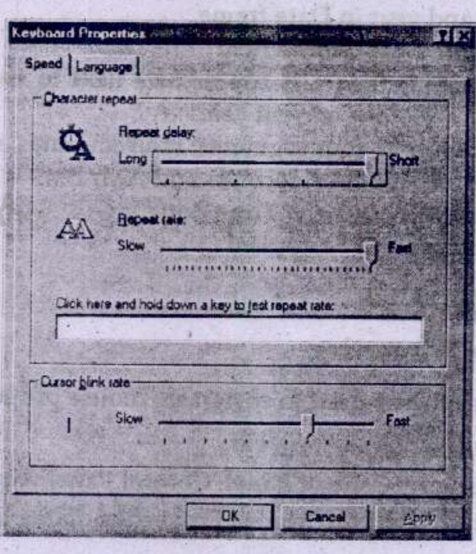
Mouse Pointer Speed and Pointer Trails

The pointer speed is the speed at which the mouse travels across the screen. The following procedure is used to adjust mouse pointer speed and trails:

1. Click **Motion** tab.
2. Adjust the slider to the desired speed level.
3. Check **Display Pointer trails** to view trails.

14.6.10 Keyboard Properties

Double click on the keyboard icon to get to the **keyboard properties**.


Keyboard Properties
Speed | Language
Character repeat
Repeat delay:
Long
Short
Repeat rate:
Slow
Fast
Click here and hold down a key to test repeat rate:
Cursor blink rate
Slow
Fast
OK
Cancel
Apply

Adjusting Keyboard Repeat Delay and Repeat Rate

The keyboard repeat delay is the amount of time between pressing a key on keyboard and seeing the character on screen. The repeat delay rate can be test by using the text box shown above. If the repeat delay is too short, the character appears as soon as the user starts holding down the key. If it is too long, it takes a second for the character to start repeating.

Adjusting Cursor Blink Rate

The cursor blink rate is the speed at which the cursor blinks on the screen. You can see the cursor blink rate to the left while making changes.

Language Tab

The Language tab is used to select a different keyboard language or keyboard layout and easily switch among languages.

14.7 Shortcut

Shortcut is used to access files, folders, drives or printers quickly. A shortcut refers to an object in the computer. The user can double click the shortcut to access the object that is referred by the shortcut. It saves a lot of time.

14.7.1 Creating Shortcuts

Different methods for creating a shortcut are as follows:

Shortcut Wizard

The following procedure is used to create a shortcut using wizard:

1. Right click on desktop. The menu will appear.
2. Select **New > Shortcut** menu. A dialog box will appear.
3. Click **Brows** button. The **Browse for Folder** dialog box will appear.
4. Select the file or folder for which the shortcut is to be created and click **OK**. The path of selected file or folder will appear in a textbox.
5. Click **Next**.
6. Enter a name for the shortcut in a textbox.
7. Click **Finish**. The shortcut icon for the selected file or folder appears on desktop.

Drag Method

The drag method is used to create a permanent shortcut on desktop. The following procedure is used to create a shortcut using drag method:

1. Right click **My Computer** on desktop. The **Quick menu** appears.
2. Select **Explore** from **Quick** menu. Windows Explorer opens.
3. In Windows Explorer, select the file or folder for which a shortcut is to be created.
4. Hold down right mouse button and drag the file or folder to desktop. The Quick menu appears.
5. Select **Create Shortcut(s) Here** from Quick menu. The shortcut icon for the selected file or folder appears on desktop.

14.8 Accessibility Features

Accessibility features in Windows 2000 Professional provide support to the users with limited sight, hearing or mobility. These features include special keyboard, sound, display and mouse configurations.

14.8.1 Accessibility Wizard

The **Accessibility wizard** is used to customize computer different tools to help the vision, hearing or mobility of special people. Accessibility options (such as StickyKeys, ShowSounds, and Mousekeys) help users with disabilities to make full use of the computer. Some of the options, such as MouseKeys, may be of interest to all users. Once the Accessibility tools are set up, they can be accessed through Control Panel and the Accessibility menu. The following procedure is used to open Accessibility Wizard:

Click **Start > Programs > Accessories > Accessibility > Accessibility Wizard.**

StickyKeys

StickyKeys is designed for people who feel difficulty to hold down two or more keys simultaneously. When a shortcut requires a key combination such as CTRL+P, StickyKeys enable the user to press a modifier key (CTRL, ALT, or SHIFT), or Windows logo key (⊞) and keep it active until another key is pressed.

The following procedure is used to turn on StickyKeys:

1. Select **Keyboard** tab.
2. Select **Use StickyKeys** check box under **StickyKeys.**

The following procedure is used to turn off StickyKeys:

1. Select **Keyboard** tab.
2. Clear **Use StickyKeys check box under StickyKeys.**

FilterKeys

FilterKeys is a keyboard feature that instructs the keyboard to ignore brief or repeated keystrokes. It can also be used to slow the rate at which a key repeats when the user holds it down. The following procedure is used to turn on FilterKeys option:

1. Open **Accessibility Options.**
2. Select **Keyboard** tab.
3. Select **Use FilterKeys** check box under **FilterKeys.**

The following procedure is used to turn off FilterKeys option:

1. Open **Accessibility Options.**
2. Select **Keyboard** tab.
3. Clear **Use FilterKeys** check box under **FilterKeys.**

ToggleKeys

ToggleKeys is designed for people with vision impairment or cognitive disabilities. When ToggleKeys is turned on, the computer provides sound cues when locking keys like CAPS LOCK, NUM LOCK or SCROLL LOCK are pressed. A high sound plays when the keys are switched on and a low sound plays when they are switched off.

The following procedure is used to turn on ToggleKeys option:

1. Open **Accessibility Options**.
2. Select **Keyboard** tab.
3. Select **Use ToggleKeys** check box under ToggleKeys.

The following procedure is used to turn off ToggleKeys option:

1. Open **Accessibility Options**.
2. Select **Keyboard** tab.
3. Clear **Use ToggleKeys** check box under ToggleKeys.

MouseKeys

MouseKeys is designed for people who have difficulty using a mouse. It allows the user to use the numeric keypad to control the mouse pointer. The MouseKeys can be activated by pressing **Num Lock to use numeric keypad** for data entry and navigation.

The following procedure is used to turn on MouseKeys option:

1. Open **Accessibility Options**.
2. Select **Mouse** tab.
3. Select **Use MouseKeys check box under MouseKeys.**

The following procedure is used to turn off MouseKeys option:

1. Open **Accessibility Options**.
2. Select **Mouse** tab.
3. Clear **Use MouseKeys** check box under **MouseKeys**.

14.9 Control Panel

Control Panel is the place to perform system management tasks. It includes installing/uninstalling new hardware devices. It is also used to manage system resources through **administrative tools**. It is also used to share printers and setup date and time etc.

14.9.1 Elements of Control Panel

Add/Remove Hardware is used to add new devices to computer. It provides a wizard that guides the user to install new hardware.

Add/Remove Programs is used to install or remove software from computer.

Data/Time element is used to adjust Date/Time and set time zone easily.

Fonts is used to install new fonts in the computer. Fonts are installed in Fonts folder of the computer.

Display element is used to control everything related to the screen. It includes colors, background, screen saver and wallpaper etc.

Game Controllers is used to play favorite game with a joystick. It is used to setup games.

Internet Options is used to change Internet settings.

Modem option is used to configure modem.

Multimedia is used for audio and video settings.

Mouse is used to configure mouse settings.

Network is related to Network/Internet.

Passwords is used to specify passwords for different users who use computer.

Power Management is used to save power. It is used to set the screen and hard drive to turn off after a specified time.

Printers is used to manager printers.

Sounds is used to set a sound for every event.

System is used to view information about the system like version of Windows, computer type: PI, PII or PIII.

Users is used to configure computer to allow different users to access system.

Regional Settings is used to configure number, currency, data and time format.

14.10 Computer Management

Computer Management is used to manage local or remote computers using a single, consolidated desktop tool. It combines several administration utilities into a single console tree. It provides easy access to a specific computer's administrative properties and tools.

The following procedure is used to open Computer Management (Local):

1. Click **Start > Control Panel**.
2. Click **Performance and Maintenance**.
3. Click **Administrative Tools** and then double click **Computer Management**.

14.10.1 Using Microsoft Management Console

Microsoft Management Console (MMC) is used to create, save and open collections of administrative tools called **consoles**. The consoles contain items such as snap-ins, extension snap-ins, monitor controls, tasks, wizards and documentation required to manage many hardware, software and networking components of Windows system.

The following procedure is used to open MMC:

- Click **Start > Run**.
- Type "mmc" in **Open** box and click **OK**.

14.10.2 Using Administrative Tools

Microsoft Management Console (MMC) hosts administrative tools to administer computers, services, other system components and networks.

Adding a Snap-in

Many administrative tools called snap-ins can be added to the console by using the following procedure:

1. Click **Start > Run**.
2. Type "mmc" in **Open** box and click **OK** to open MMC.
3. Click **File > Add/Remove Snap-in**. The Add/Remove Snap-in dialog box appears.
4. Click **Add**. The Add Standalone Snap-in dialog box appears.
5. Click the snap-in to be added to the console and click Add.

14.11 Using Printers with Windows

A printer can either be directly or remotely connected to your computer. A printer physically connected to your computer via a cable is called a local printer.

A printer physically attached to another computer and available for use by your computer through your network connection is called a network printer.

14.11.1 Installation of Local Printer

The procedure for adding a new printer is as follows:

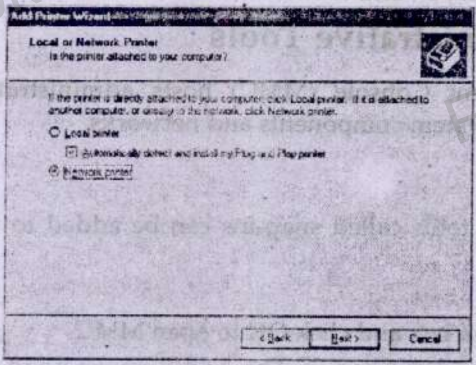
1. Select **Start > Setting > Printers**. The Printers Window will open.
2. Click **Add Printer** icon in **Printers** to launch Add New Printer wizard.
3. Click **Next** button to begin printer installation process.
4. Select **Local Printer** radio button.

5. Click **Next** button.
6. Select the port to which the printer is connected in **Available Ports** list.
7. Click **Next**.
8. Scroll down in **Manufacturers** list and select the manufacturer of printer. It displays list of printer models.
9. Select the model of printer and click **Next** button. If printer is not listed, click **Have Disk** button and browse to the location of driver files.
10. Enter name of the printer. By default, the system uses the name associated with the printer driver.
11. Click **Next. Print Test Page** will appear.
12. Select **Yes** to print a test page or **No** to skip test page.
13. Click **Finish** button to complete the installation.

14.11.2 Connecting to a Network Printer

The network name of the printer is required to connect to the network printer. The following procedure is used to connect to the network printer:

1. Select **Start > Settings > Printers**. The Printers window appears.
2. Double click **Add Printer** icon. The **Add Printer Wizard** dialog box appears.
3. Click **Next**.

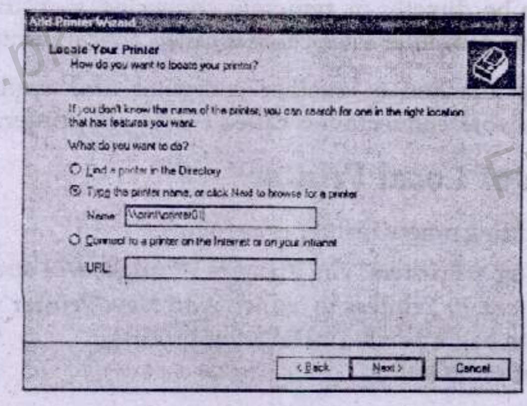


Local or Network Printer
Is the printer attached to your computer?

If the printer is directly attached to your computer, click Local printer. If it is attached to another computer, or directly to the network, click Network printer.

○ Local printer
☑ Automatically detect and install my Plug and Play printer
● Network printer

4. Select Network printer and click Next
5. Select **Type the printer name, or click Next to browse for a printer.**
6. Type the name of printer in Name text box and click **Next**. The printer names should be entered in the following format: \\print\printer-name.
   For example \\print\print01



Locate Your Printer
How do you want to locate your printer?

If you don't know the name of the printer, you can search for one in the right location that has features you want.

What do you want to do?

○ Find a printer in the Directory
● Type the printer name, or click Next to browse for a printer
Name: \\print\printer01
○ Connect to a printer on the Internet or on your intranet
URL:

7. Click **Next**.
8. Verify that printer name and default setting is correct. If the information is incorrect, click **Back** and repeat the necessary steps. If the information is correct, click **Finish**. The network printer is added.

14.11.3 Managing Printer Queue

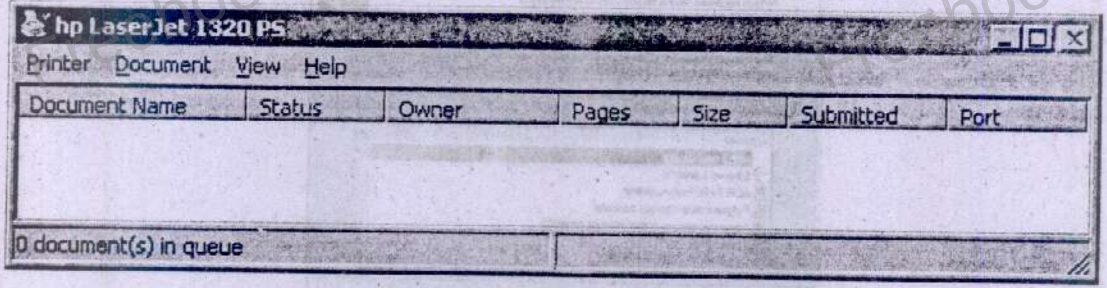
Print queue is a collection of all documents that are waiting for printing task. Windows maintains a print queue for all print jobs. Print queue can be used to restart or cancel and printing task.

Print queue is used because the speed of printer is slower than computer. If multiple printing requests are issued, the printer cannot print them all at once. That is why print queue is used. The print jobs are printed in the same sequence in which they are received in the print queue.

Print queue is displayed in print queue dialog box. This dialog box can be displayed in the following two ways:

- Click **Start** menu.
- Click **Printers** menu and then double click on printer in **Printers** folder.

A printer icon is displayed in system tray when the printer is printing a document. The user can double click on the icon to view print queue..



0 document(s) in queue

14.12 Adding and Removing Hardware

Installing a plug and play device in Window 2000 is very easy. Windows 2000 installs all necessary drivers, updates the systems and allocates resources while installing a hardware device.

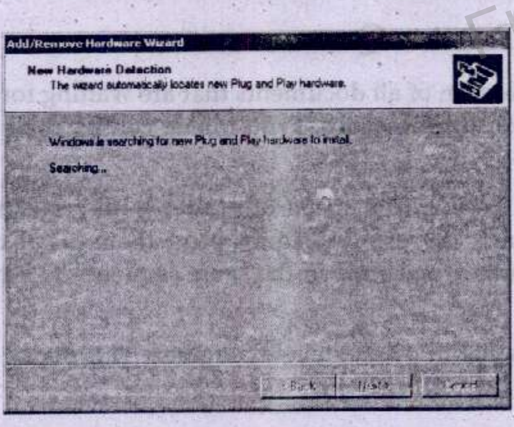
14.12.1 Adding Null Modem

The **Add/Remove Hardware** wizard is used to install a Null modem. Null modem is a concept to install the required drivers without the physical presence of a modem.

The following procedure is used to add a Null modem:

1. Click **Start > Setting > Control Panel**. A control panel window will appear.
2. Double click **Add/Remove Hardware**. The **Add/Remove Hardware** wizard will appear.
3. Click **Next**.
4. Select **Add/Troubleshoot a device** radio button.
5. Click **Next**.

The **Add/Troubleshoot a device** radio button is used to add or troubleshoot hardware device. The **Uninstall/Unplug a device** radio button is used to uninstall hardware. The wizard searches for the existing hardware devices in the system. It displays a list of existing hardware devices.



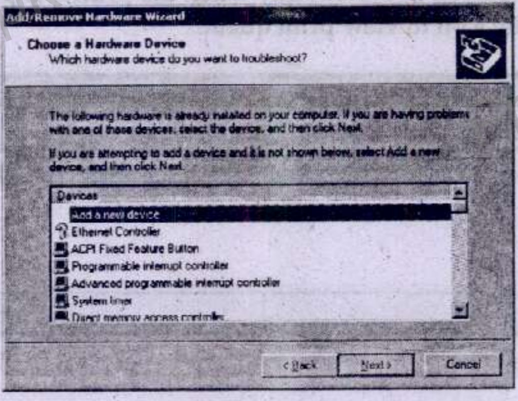
Add/Remove Hardware Wizard
New Hardware Detection
< Back
Next >
Cancel

The wizard automatically locates new Plug and Play hardware.

Windows is searching for new Plug and Play hardware to install.

Searching...

6. Select **Add a new device** option from the devices list.
7. Click **Next** button.



Add/Remove Hardware Wizard
Choose a Hardware Device
Devices
< Back
Next >
Cancel

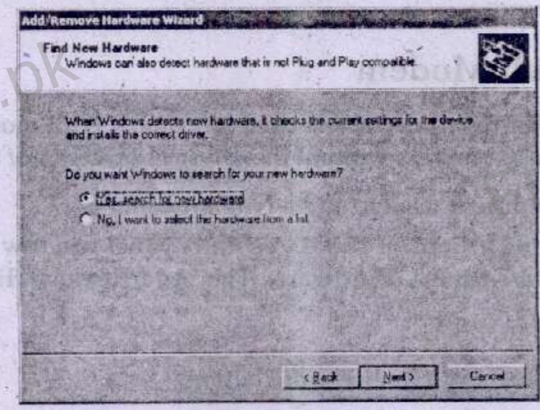
Which hardware device do you want to troubleshoot?

The following hardware is already installed on your computer. If you are having problems with one of these devices, select the device, and then click Next.

If you are attempting to add a device and it is not shown below, select Add a new device, and then click Next.

- Add a new device
- Ethernet Controller
- ACPI Fixed Feature Button
- Programmable interrupt controller
- Advanced programmable interrupt controller
- System timer
- Direct memory access controller

8. Select the **Yes, search for new hardware** radio button and click **Next** button. The wizard automatically detects the new hardware devices.



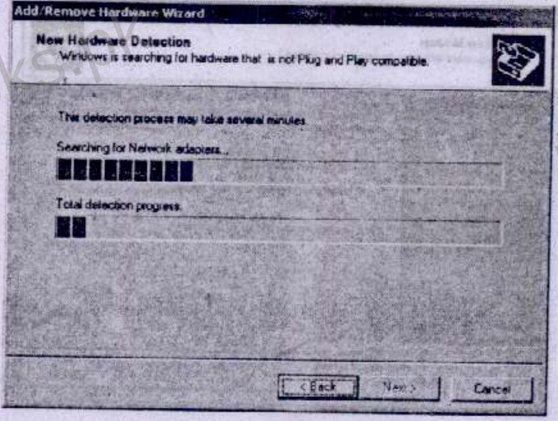
Add/Remove Hardware Wizard
Find New Hardware
< Back
Next >
Cancel

Windows can also detect hardware that is not Plug and Play compatible.

When Windows detects new hardware, it checks the current settings for the device and installs the correct driver.

Do you want Windows to search for your new hardware?

● Yes, search for new hardware
○ No, I want to select the hardware from a list


Add/Remove Hardware Wizard
New Hardware Detection
< Back
Next >
Cancel

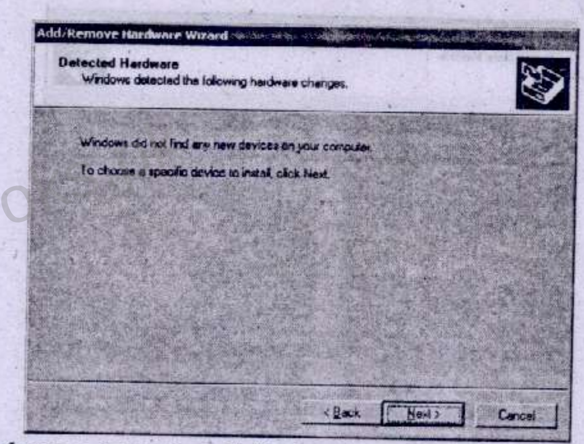
Windows is searching for hardware that is not Plug and Play compatible.

This detection process may take several minutes.

Searching for Network adapters...

Total detection progress

9. As no hardware device is attached to the system, the following figure will be shown.

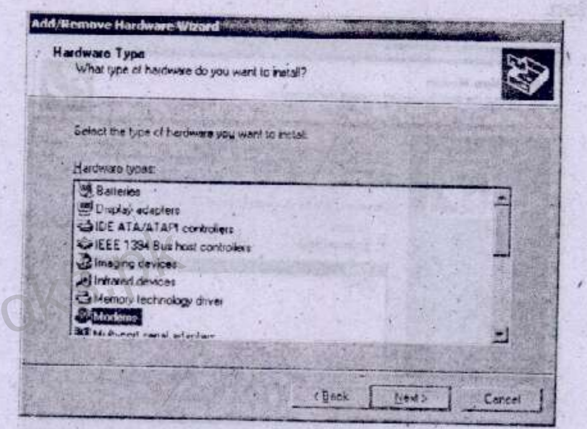

Add/Remove Hardware Wizard
Detected Hardware
< Back
Next >
Cancel

Windows detected the following hardware changes.

Windows did not find any new devices on your computer.

To choose a specific device to install, click Next.

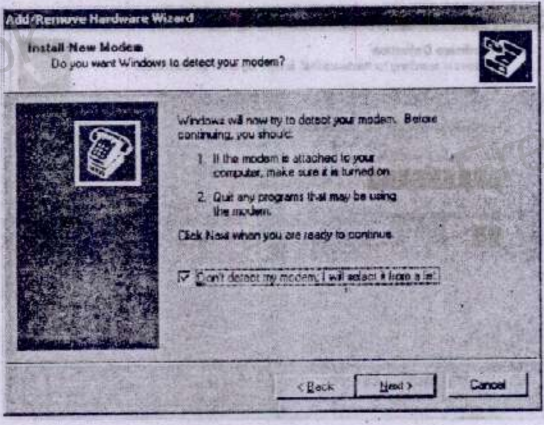
10. Click on Next button. List of available hardware devices will be shown.


Add/Remove Hardware Wizard
Hardware Type
Hardware types:
Batteries
Display adapters
IDE ATA/ATAPI controllers
IEEE 1394 Bus host controllers
Imaging devices
Infrared devices
Memory technology driver
Modems
< Back
Next >
Cancel

What type of hardware do you want to install?

Select the type of hardware you want to install.

11. Select **Modems** from the hardware list.
12. Click on **Next** button. If modem is attached to computer, the wizard automatically detects it and configures the system properly. Since there is no modem attached to the system, the following screen appears;



Add/Remove Hardware Wizard

Install New Modem

Do you want Windows to detect your modem?

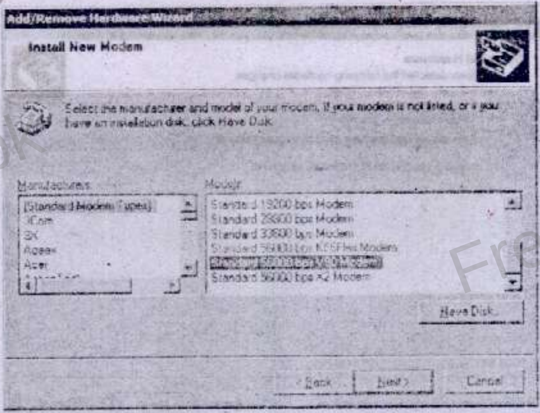
Windows will now try to detect your modem. Before continuing, you should:

1. If the modem is attached to your computer, make sure it is turned on.
2. Quit any programs that may be using the modem.

Click Next when you are ready to continue.

☑ Don't detect my modem; I will select it from a list.

13. Select **Don't detect my modem. I will select it from a list checkbox** and click **Next**.



Add/Remove Hardware Wizard

Install New Modem

Select the manufacturer and model of your modem. If your modem is not listed, or if you have an installation disk, click Have Disk.

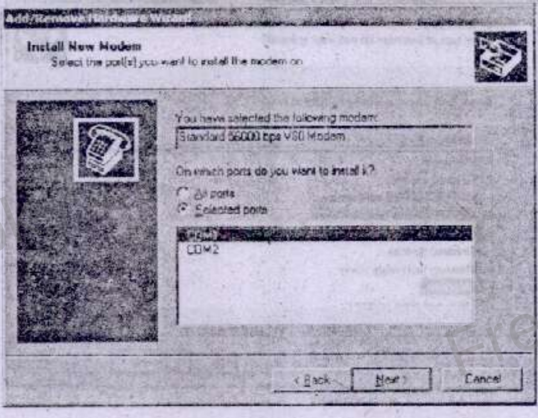
Manufactures:

(Standard Modem Types)
3Com
3X
Apex
Acer

Models:

Standard 19200 bps Modem
Standard 28800 bps Modem
Standard 33600 bps Modem
Standard 56000 bps K56Flex Modem
[illegible]
Standard 56000 bps X2 Modem

14. Select **Manufactures** and **Models** as shown in figure.
15. Click **Next** button.



Add/Remove Hardware Wizard

Install New Modem

Select the port(s) you want to install the modem on.

You have selected the following modem:

Standard 56000 bps V90 Modem

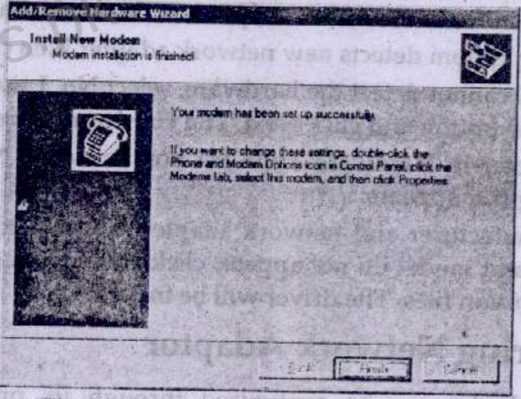
On which ports do you want to install it?

○ All ports
● Selected ports

[illegible]
COM2

16. Select **Selected ports** radio button and select **COM1** port and click **Next** button.

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

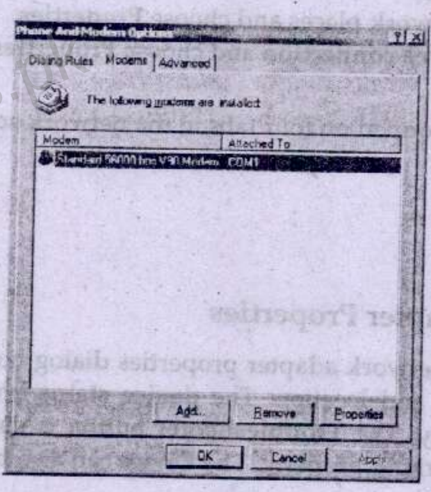


Modem installation is finished

Your modem has been set up successfully.

If you want to change these settings, double-click the Phone and Modem Options icon in Control Panel, click the Modems tab, select this modem, and then click Properties.

17. Click **Finish** button. The Null modem is successfully installed.
18. Select **Start** > Setting > Control Panel > Phone and Modem.
19. Select **Modems** tab on **Phone and Modem** options window. The modem is installed.
20. Close the **Phone and Modem** options window.



The following modems are installed:

Modem	Attached To
Standard 56000 bps V90 Modem	COM1

14.12.2 Installing a Network Adapter

The user should read the instructions that come with hardware before installing the Network adapter. The network adapter should be self configuring with plug-and play if it is turned on. If the network adapter is not Plug-and-Play, the operating system should detect it after it has been installed and start a Wizard to load adapter's driver.

The following procedure is used to install a driver for the network adapter that is not pug-and-Play:

1. Select **Start > Control Panel > Add/ Remove Hardware**. The **Welcome to Add/Remove Hardware Wizard** dialog box starts and **Choose a Hardware task** dialog box appears.
2. Select **Add/Trouble Shoot a Device** radio button and click **Next**. Windows 2000 detects new hardware and displays **Choose a Hardware Device** dialog box.
3. Choose **Add a new device** option at the top of **Device** list box and click **Next**. The **Find new hardware** dialog box appears.

4. Select **Yes, search for new hardware** option to detect the hardware.
5. If the operating system detects new network adapter then click **Next**.
6. If window 2000 cannot detect the hardware, select **No, I want to select the Hardware from a list** radio button and click **Next**. The **Hardware type** dialog box appears.
7. Select the adapter in **Hardware Type** list box and click **Next**. The **Select Network Adapter** dialog box appears.
8. Select the manufacturer and network adapter model if it appears in the list. If the manufacturer and model do not appear, click **Have Disk** and specify the location of driver's distribution files. The driver will be installed.

14.12.3 Configuring Network Adaptor

The network adapter can be configured through its properties dialog box after installing it. The following procedure is used to access properties dialog box:

1. Select **Start > Settings > Control Panel > Network and Dial-up Connections > Local Area Connection > Properties**.
2. Click **Configure**. **OR**

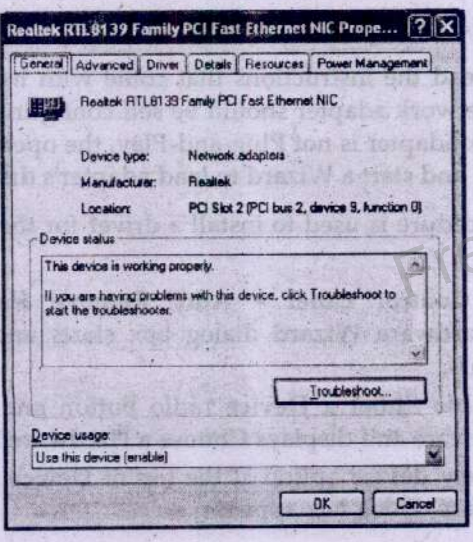
1. Right-click **My Network** places and choose **Properties**.
2. Right-click **Local area connection** and choose **Properties**.
3. Click **Configure**.

The properties are grouped on four tabs in the network adapter properties dialog box:

- General
- Advanced
- Driver
- Resources

General Network Adapter Properties

The **general** tab of network adapter properties dialog box shows the name of adapter, device type, manufacture and location. The **device status** box reports whether or not the device is working properly. The **Troubleshooter** button is used to display troubleshooting tips if the device is not working properly. The device can also be enabled or **disabled** through **the device** drop-down list.



Realtek RTL8139 Family PCI Fast Ethernet NIC

Device type: Network adapters

Manufacturer: Realtek

Location: PCI Slot 2 (PCI bus 2, device 9, function 0)

Device status

This device is working properly.

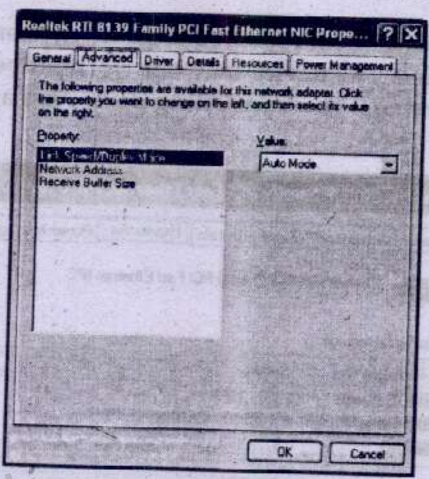
If you are having problems with this device, click Troubleshoot to start the troubleshooter.

Device usage:

Use this device (enable)

Advanced Network Adapter Properties

The contents of the advanced tab vary depending on the network adapter and driver. The following figure shows an example of advanced tab for a fast Ethernet adapter.


Realtek RTl 8139 Family PCI Fast Ethernet NIC Prope...
?
X
General
Advanced
Driver
Details
Resources
Power Management
OK
Cancel

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:
Link Speed/Duplex Mode
Network Address
Receive Buffer Size

Value: Auto Mode

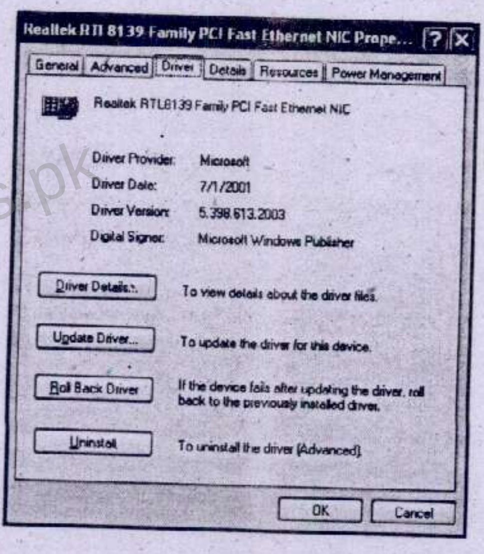
Driver Properties

The **driver tab** provides the following information about the driver:

- The driver provider which is usually Microsoft or network adapter manufacture.
- The date of release of driver
- The driver version
- The digital signer. It is the company that provides digital signature.

The **Driver Details** button at the bottom of **Driver** tab displays the following details about the driver.

- The location of driver file that is useful for troubleshooting.
- The original provider of driver that is usually the manufacture.
- The file version that is useful for troubleshooting.
- The copyright information about the driver.


Realtek RTl 8139 Family PCI Fast Ethernet NIC Prope...
?
X
General
Advanced
Driver
Details
Resources
Power Management
Driver Details...
Update Driver...
Roll Back Driver
Uninstall
OK
Cancel

Realtek RTL8139 Family PCI Fast Ethernet NIC

Driver Provider: Microsoft
Driver Date: 7/1/2001
Driver Version: 5.398.613.2003
Digital Signer: Microsoft Windows Publisher

To view details about the driver files.

To update the driver for this device.

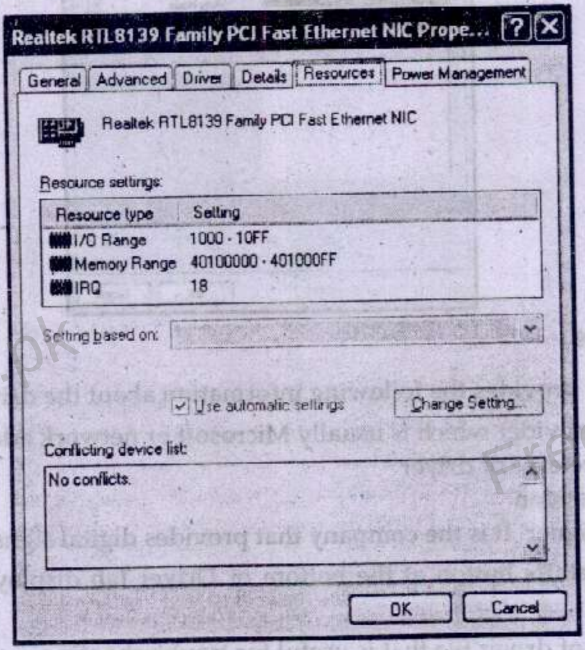
If the device fails after updating the driver, roll back to the previously installed driver.

To uninstall the driver (Advanced).

The **Uninstall** button at the bottom of **Driver** tab removes the driver form computer. The **Update Driver** button is used to update the driver.

Resource Properties

The **resource** tab displays the resource settings for network adapter. The information is important for troubleshooting. The device may not work propery if other devices are trying to use the same resource settings. The **Conflicting device list** at the bottom of **Resources** tab shows if any conflicts exist.


Realtek RTL8139 Family PCI Fast Ethernet NIC Prope...
?
General
Advanced
Driver
Details
Resources
Power Management
Change Setting...
OK
Cancel

Realtek RTL8139 Family PCI Fast Ethernet NIC

Resource settings:

Resource type	Setting
I/O Range	1000 - 10FF
Memory Range	40100000 - 401000FF
IRQ	18

Setting based on:

☑ Use automatic settings

Conflicting device list:

No conflicts.

15 MANAGING USERS, FILES & FOLDERS

Chapter Overview

15.1 User Accounts
    15.1.1 Types of User Accounts
    15.1.2 Creating User Account
15.2 Groups
    15.2.1 Local Groups
    15.2.2 Built-in Groups
15.3 Auditing
    15.3.1 Audit Policy
    15.3.2 Planning an Audit Policy
    15.3.3 Implementing Audit Policy
    15.3.4 Setting an Audit Policy
15.4 Account Policies
    15.4.1 Password Policy
    15.4.2 Account Lockout Policy
15.5 Security Options
15.6 Group Policy Object and Active Directory
    15.6.1 Active Directory
15.7 Profiles
    15.7.1 Types of Profiles
    15.7.2 Logon Script
    15.7.3 Setting up Home Folder
15.8 File and Folder Management
    15.8.1 Organizing Files and Folders
    15.8.2 Managing Folder Options
    15.8.3 Searching for Files and Folders
15.9 Shared Folders
    15.9.1 Creating a Shared Folder
    15.9.2 Shared Folder Permissions

Review Questions

15.1 User Accounts

A user account is a record that is used to store user information of a network. It is normally stored locally and on the server of network. It consists of user name, password and logon restrictions for the user. Each user has a user account. It is activated when the user logs on to the computer or network.

15.1.1 Types of User Accounts

Windows 2000 provides three different types of user accounts:

1. Local User Account

The local user accounts provide access to the resources on local commuter. Local user accounts are created only on computers that are not in a domain. Windows 2000 creates the local user account only in that computer's security database called **local security database**. Windows 2000 does not replicate local user account information to any other computer.

2. Domain User Accounts

A domain user account allows a user to logon to a domain to access network resources. The user provides password and user name during logon process. Windows 2000 uses this information to authenticates the user. It then builds an **access token** that contains information about user and security setting. The domain user accounts can only be created on domain.

3. Built-in User Accounts

A built-in user account allows a user to perform administrative tasks and gain access to local or network resources. Windows 2000 automatically creates built-in accounts. Two commonly used built-in accounts are administrator and guest.

- **Administrator Account:** A person who controls this account is called **administrator** of the network. He creates other network users accounts.
- **Guest Account:** The guest account is designed for the occasional user. The person who does not require regular use of the network can log in through guest account. For example, a person can access the printer on another network using this account. The guest account offers limited access to resources on the network.

15.1.2 Creating User Account

The **computer management** tools are used to create local user accounts. The following procedure is used to create user account:

1. Select **Start > Programs > Administrative Tools > Computer Management**.
2. Click **Local Users** in the computer management window in the consol pane.
3. Right click on **User Folder** in detail pane. A pop-up menu appears.
4. Click on **New User**.
5. Fill in the appropriate fields in the new user dialog box
6. Click on **Create** button. The user account will be created.

Local User Account Options

Some options of local user account are as follows:

- **User Name:** The users logon name. This field is required.

- **Full Name:** The users full name. It consists of first name and last name. It can also include the middle name or initial.
- **Description:** An optional field to type descriptive text about user account or the user.
- **Password:** The password is used to authenticate the user. It is represented as asterisk.
- **Confirm password:** It confirms the password by typing it second time to make sure that the user typed the password correctly. It is required if a password is used.
- **User must change password at next logon:** It is used to require the user to change the password at fist logon. It ensures that the user is the only person who knows the password. It is selected by default.
- **User cannot change password:** It is used many person use the same user account such as guest. It allows only administrators to control password. This option is not available if the **user must change password at next logon** check box is selected.
- **Password never expires:** It allows the user to use the same password forever.
- **Account is disabled:** It is used to prevent user from using this user account.

Practical

Create the following user accounts and test the log on procedure with one of the users.

Username	Full name	Password	Change password
User1	User one	Password	Must
User2	User Two	Password	Blank
User3	User three	User3	Must
User4	User four	User4	Blank

Procedure

1. Log on as administrator.
2. Select **Start > Programs > Administrator tools > Computer Management**. The **Computer Management** window will appear.
3. Click **Local Users and Groups**.
4. Right click **Users** and select **New User**. The **New User** dialog box will appear.
5. Type "user1" in **User Name** box.
6. Type "user one" in **Full Name** box.
7. Type "password" in **Password** and **Confirm Password** fields.
8. Specify whether or not the user can change the password.
9. Click **Create** after selecting the appropriate password options. The **New User** dialog box clears and remains displayed so that you can create another user account.
10. Complete steps 5 - 9 for the remaining users.
11. Click **Close** to close **New User** dialog box.
12. Close **Computer Management** window.

Test Local User Account

1. Log off as administrator.
2. Logon on as user 1 with a password of "password".
3. Enter new password in New Password and Confirm Password fields when a dialog box appears.
4. Click OK.

Disable a User

1. Expand **Local Users and Group** in **Computer Management** window.

2. Open **User** folders.
- Double click user1 to open properties dialog box
3. Select **General** tab and check **Account Is Disable** box.
4. Click **OK**.
- Log off as administrator and attempt to logon on as user1. The attempt will fail as the account is disabled.

Delete User Account

1. Log on as administrator.
2. Expand **Local Users and Group in Computer Management** window.
3. Open **User** folders.
- Click on **user1** to select the user account
4. Press **Delete**. A dialog box will appear to confirm user deletion.
5. Click **Yes** to confirm the deletion.

Rename a User

1. Log on as administrator.
2. Expand **Local Users and Group** in **Computer Management** window.
3. Open **User folders**.
4. Right click user2 and select **Rename**.
5. Type username **user22** and press **Enter**. The user name will be changed.

Changing Password

1. Log on as administrator.
2. Expand **Local Users and Group in Computer Management** window.
3. Open **User** folders.
4. Right click user22 and select **Set Password**. The **Set Password** dialog box appears.
5. Type the new password and then confirm the password.
6. Click **OK**. The password will be changed.

Setting Properties for User Account

A set of default properties is associated with each local user account. These properties can be configured after creating the account. A **User Properties** dialog box has three tabs that contain information about each user account.

- General tab
- Member of tab
- Profile tab

Practical: To Test "User Must Change Password at Next Login" Property

1. Log on as **user3**. Windows 2000 displays a **Logon Message dialog box** to change password at first logon.
2. Click **OK**. Windows 2000 displays a **Change Password** dialog box.
3. Type password in **New Password** and **Confirm New Password** fields.
4. Click **OK**. A dialog box will appear to indicate that password has been changed.
5. Click **OK**.

Practical: To Test "User cannot Change Password" Property

The following example will set and test that user cannot change password property.

1. Log on administrator.

2. Start **Computer Management** from **Administrator Tools** menu.
3. Expand **Local User and Groups** folder and click **Users**. Windows 2000 displays the users in the detail pan.
4. Right-click on **user1** and click **Properties**. The **User1 Properties** dialog box appears.
5. Select **User Cannot Change Password** checkbox.
6. Click **OK** to close user1 Properties dialog box.
7. Right-click on **user2** and select **Properties**. The **user2 Properties** dialog box appears.
8. Select **Account Is Disable** checkbox.
9. Click **OK** to close user2 **Properties** dialog box.
10. Close **Computer Management** and log off the computer.

15.2 Groups

A **group** is a collection of user accounts. All members of a group can be assigned rights and permissions at the same time. Any member added to the group automatically gets rights and permissions of the group. Only users with administrative rights can create groups.

The groups are created to simplify account administration. An administrator can assign rights and permission to individual users if the network is small. It is very difficult to assign rights and permissions to a large number of users. The groups are created on basis of common functionality or location.

15.2.1 Local Groups

A **local group** is a collection of user accounts on a computer. The members of local groups have permissions to the resources on the computer on which the group is created. Windows 2000 creates local group in the local security database. It can not be created on domain controller. Local groups cannot be member of any other group.

Creating Local Groups

The following procedure is used to create local groups:

1. Expand **Local Users and Groups** in **Computer Management** window.
2. Click **Groups** folder.
3. Right-click **Groups** and click **New Group**. The **New Group** dialog box appears with the following options:

Option	Description
Group Name	A unique name for local group. It is compulsory entry. It may consist of up to 256 characters.
Description	A description of the group
Add	Adds a user to the list of members
Remove	Removes a user from the list of members
Create	Creates the group
Close	Closes the new group dialog box

4. Enter the appropriate information and click Create.

Deleting Local Groups

The **Computer management** window is used to delete local groups. When a group is deleted, only the group, permissions and rights of the group are deleted. It does not delete user account of the group.

- Right click the group and click **Delete**.

Adding Members to a Group

The following procedure is used to add a member to a group:

1. Start **Computer Management**.
2. Expand **Local Users and Groups**.
3. Click **Groups** and then the detail panes.
4. Right click the appropriate group and click **Properties**.
5. Click **Add** in **Properties** dialog box. The selected users or groups dialog box appears.
6. Ensure that the computer on which you created the group is selected in **look in list**.
7. Select the user account that to be added to the group in **Name** box.
8. Click **Add**.

Practical

Create two local groups **You and Me**. **Add members to both groups.**

1. Log on administrator.
2. Click **Start > Programs > Administrator tools > Computer Management**.
3. Expand **Local Users and Groups** and click **Groups**. **The Computer Management** displays a list of current and built-in local groups in the detail panes.
4. Right click **Groups** and click **New Group**. The **New Group** dialog box will appear.
5. Type "You" in **Group Name** box and "access to customer files" in **Description** box.
6. Click **Add**. The **Select users or groups** dialog box appears.
7. Hold the Ctrl key and select user1 and user3.
8. Click **Add**. The User1 and user3 will appear in box below **Add** button.
9. Click **OK**. The user1 and user3 are listed in **member** box.
10. Click **Create**. Windows will create group and add it to the list of users and groups.
11. Repeat steps 4 – 10 to create a group "Me".

Add and Remove Members from Local Group

1. Double click "You" in the detail pane of **Computer Management**. The **Properties** dialog box will appear. The user2 and user4 will appear in **Members** box.
2. Click **Add** to add a member to the group. The **select users or groups** dialog box will appear.
3. Select user3 in **Name** box.
4. Click **Add** and click **OK**. The "You" properties dialog box will display user2, user3 and user4 in **Member** box.
5. Select user4 and click **Remove**. The user4 will disappear from **Member** box. It still exits as local user account but it is not a member of "You" group.
6. Click **OK**.

Renaming Groups

1. Start Computer **Management** and expand **Local Users and Groups**
2. Expand **Group** folder.
3. Right click the group "You" and select **rename**.
4. Rename the group to "New" and press **Enter**.

Deleting a Group

- Right click the group "New" in **Computer Management** details pane.
- Click **Delete**. A local users and groups dialog box appears to confirm the deletion.

- Click **Yes**. The group "New" will disappear. The members of the group are not deleted. The User2 and user 3 are still local user accounts.
- Close Computer Management.

15.2.2 Built-in Groups

All computers running Windows 2000 Professional have built-in local groups. The built-in local groups give rights to perform system taks on a single computer such as:

- Backing up and restoring files
- Changing the system time
- Administering system resources

Windows 2000 places built-in local group in groups folder in computer management.

Group	Description
Users	It can perform tasks according to the given rights and can access permitted resources.
Administrators	It can perform all administrative tasks.
Guests	It can perform tasks according to the given rights and can access permitted resources. No permanent changes can be made to the local environment.
Back up operator	Use Windows 2000 backup to backup and restore all computers running Windows 2000
Replicator	To be used by the directory replicated service

Built-in System Groups

Built-in in system groups exists on all computers running windows 2000. System groups do not have specific membership that cannot be modified. They can represent different users at different times depending on how users gain access to group. They are vailable for the user when rights and permission to resources are assigned.

Group	Description
Everyone	All local and remote uses. Any rights or permission can be assigned to this group. It contains user accounts other than created by an administrator.
Creator owner	It includes owner of a resource. It is used only on NTFS volumes.
Network	It includes any user connected to computer to access network.
Interactive	It includes any user who logs on locally to a computer.

5.3 Auditing

Auditing is used to track the activities of user and Windows 2000. These activities are lled events. The user can specify through auditing that Windows 2000 writes a record of an vent to the security log. The security log maintains a record of valid and invalid logon tempts and events related to createing, opening or deleting files or other objects.

An audit entry in security log contains the following information:

- The action that was performed
- The user who performed the action
- The success or failure of the event and when the event occurred

Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk Freebooks.pk

15.3.1 Audit Policy

An audit policy defines the types of security events that Windows 2000 records in the security log on each computer. The security log is used to track the events specified by user.

Windows 2000 writes events to security log on the computer where the event occurs For example, if a user attempts to log on and the attempt fails, Windows 2000 writes an even to security log on the computer.

An audit policy can be set up for a computer to do the following:

- Track the success and failure of events such as:
- Logon attempts by users
- An attempt by a particular user to read a specific file
- Changes to a user account or group membership
- Changes to security settings.
- Eliminate or minimize the risk of unauthorized use of resources.

15.3.2 Planning an Audit Policy

The following things must be determined while planning an audit policy:

- What is to be audited
- Which computer is to be used to configure auditing

The following types of events can be audited:

- Accessing files and folders
- Logging on and off
- Shutting down and restarting a computer running Windows 2000
- Changing user accounts and groups
- Attempting to make changes to objects in directory services based on Acti Directory technology (if Windows 2000 computer is part of a domain)

The user must determine the types of events to audit. He must also determine wheth to audit the success of events, failure of events or both. Tracking successful events can tell t user how often Windows 2000 or users access specific files, printer or other objects. T information can be used for resource planning.

Tracking failed events can alert the user about security. For example, if there are ma failed log on attempts for a particular user, it can alert the user that some unauthoriz person might be attempting to access the computer.

Some other guidelines for planning an audit policy are as follows:

- The user should determine whether the tracking of trends of system use is needed it is needed, he should plan to archive event logs. Archiving the logs will allow u to view how system use changes over time. It also allows him to plan to increa system resources if needed.
- The user should review security logs frequently. It helps to find out any secur problems.
- The user should define an audit policy that is useful and manageable. Only sensit and confidential data should be audited. Only those events should be audited t provide meaningful information about network environment. Auditing too ma events can create excess overhead for Windows 2000.

- The user should audit resource access by using **Everyone** group instead of **Users** group. It ensures that everyone who connects to the network is audited.

15.3.3 Implementing Audit Policy

An audit policy for each individual computer is set up for the computer running Windows 2000 Professional.

Auditing Requirements

The requirements to set up and administer auditing are as follows:

- You must have **Manage Auditing And Security Log** user right for the computer where an audit policy is to be configured or reviewed. Windows 2000 grants these rights to the **Administrators** group by default.
- The files and folders to be audited must be on Microsoft Windows 2000 File System (NTFS) volumes.

Setting up Auditing

Setting up an auditing consists of the following two parts:

- **Set the audit policy:** The audit policy enables auditing of objects but it does not activate auditing of specific objects.
- **Enable auditing of specific resources:** The user specifies the specific events to audit for files, folders, printers and Active Directory objects. Windows 2000 then tracks and logs the specified events.

15.3.4 Setting an Audit Policy

The first step in implementing an audit policy is selecting the types of events that Windows 2000 audits. The audit policies can be set in **Local Security Setting** window. It can be opened from **Administrative Tools > Local Security Policy**.

The following table describes the types of events that Windows 2000 can audit:

Event	Description
Account Logon Events	A domain controller received a request to validate a user account. It is applicable only if the computer running Windows 2000 joins a Windows 2000 domain.
Account Management	An administrator created, changed or deleted a user account or group. A user account was renamed, disabled or enabled or password was set or changed.
Directory Service Access	A user gained access to an Active Directory object.
Logon Events	A user logged on or logged off or a user made or canceled a network connection.
Object Access	A user gained access to a file, folder or printer.
Policy Change	A change was made to the user security options, user rights or audit policies.
Privilege Use	A user exercised a right such as changing the system time etc.
Process Tracking	A program performed an action.
System Events	A user restarted or shut down the computer or any event occurred that affects Windows 2000 security or the security log.

15.4 Account Policies

The account policies include password policy and system lockout policy.

15.4.1 Password Policy

Password policy is used to improve the security on the computer. It controls how passwords are created and managed. You can specify the maximum length of time a password can be used before the user must change it. If the password is changed frequently, it decreases the chance unauthorized use of computer by any person.

You can also specify a minimum password length. A longer password is more difficult to discover. You can also maintain a history of passwords that are used.

Password Policy on a computer running Windows 2000 Professional can be configured by using **Group Policy** or **Local Security Policy**. Group Policy is used to configure Password Policy as follows:

1. Use **MMC** to create a custom console, add **Group Policy** snap-in and save it with **Group Policy** name
2. Use **Local Computer Policy > Computer Configuration > Windows Setting > Security Setting > Account Policies > Password Policy.**
3. Select the setting to be configured and then click **Security** on **Action** menu.

The following tables describes the settings available in Password Policy:

Setting	Description
Enforce Password History	It indicates the number of passwords to be kept in password history. The value can be from 0 to 24. The default value is 0.
Maximum Password Age	It indicates the number of days a user can access a password before he is required to change it. The value can be from 0 to 999 days. The default value is 42 days.
Minimum Password Age	It indicates the number of days a user must keep a password before he can change it. The value can be from 0 to 999 days.
Minimum Password Length	It indicates the minimum number of characters required in a password. It can be from 0 to 14 characters.
Passwords Must Meet Complexity Requirements	It can be Enabled or Disabled. The default value is Disabled.
Store Password Using Reversible Encryption For All Users In The Domain	It can be Enabled or Disabled. The default value is Disabled.

15.4.2 Account Lockout Policy

Account Lockout Policy is used to improve the security on the computer. If this policy is not used, an unauthorized user can attempt to break into the computer repeatedly. If the policy is used, the system will lock out the user account after specified number of failed attempted. Account Lockout Policy can be accessed by using Group Policy snap-in or Local Security Setting window.

The following table describes the settings available in Account Lockout Policy:

Setting	Description
Account Lockout Duration	It indicates the number of minutes that the account is locked out. The 0 value indicates that account is locked out indefinitely until Administrator unlocks it. The value can be from 0 to 99999 minutes.
Account Lockout Threshold	It indicates the number of invalid logon attempts it takes before the user account is locked out. A 0 value indicates that account will not be locked out. The value can be from 0 to 999 attempts.
Reset Account Lockout Counter After	It indicates number of minutes to wait before resetting account lockout counter. The value can be from 1 to 99999 minutes.

15.5 Security Options

The **Security Options** lies under **Local Policies**. It is used to increase effective security on the computer.

Shutting Down Computer without Logging On

Windows 2000 Professional does not require to log on to shut down the computer by default. Security Option can be used to disable this option and force the user to log on before shutting down the computer. Security Option can be accessed using Group Policy snap-in.

Clear Virtual Memory Pagefile when System Shuts Down

By default, Windows 2000 Professional does not clear the virtual memory pagefile when the system is shut down. The data in pagefile might be accessible to the users who are not authorized to view it. The following procedure can be used to enable this feature:

Disable CTRL+ALT+DEL Requirement for Logon

By default, Windows 2000 Professional does not require users to press Ctrl+Del+Del to log on. This feature can be disabled to increase the security.

Do Not Display Last User Name in Logon Screen

By default, Windows 2000 Professional displays the last user name to log on in **Windows Security** or **Log On To Windows** dialog box. It is considered a risk in some cases as an unauthorized person can see a valid user account. It becomes easier for him to break into the system.

15.6 Group Policy Object and Active Directory

Group Policy Objects are normally applied through Active Directory. It is easier to manage LGPOs globally than apply them at local levels.

15.6.1 Active Directory

Active **Directory** provides several levels of hierarchical structure. The **domain** is the main unit of organization in Acive Directory. A domain may include many objects such as users, groups and GPOs. Security can be applied on each object in the domain. The security specifies who can access the ojbect. It also specifiy the level of access of different users. The domain objects can be further subdivided and organized using **Organizational Units**.

Practical: Setting up a Password Policy

Create a password policy and then logon with a user account to test that policy.

Procedure

1. Log on as **Administrator**.
2. Select **Start > Programs > Administrative Tools > Local Security Policy**.
3. Click on plus next to **Local Users and Groups** entry.
4. Right click on **Users** folder in left hand windowpane. A popup menu appears.
5. Click on **New User** option. The **New User window** appears.
6. Enter "Usman" in **username** field.
7. Enter "Usman Khalil" in **Full Name** field.
8. Enter "Manager" in **Description** field.
9. Enter "Password" in **Password** and a **Confirm Password** fields.
10. Clear tick box **User Must Change Password at Next logon**.
11. Click on **Create** button. The user will be created and the **New User** window appears.
12. Enter "Abdullah" in **user name** field.
13. Enter "Muhammad Abdullah" in **Full Name field**.
14. Enter "Security Advisor" in **Description** field.
15. Enter "password" in **Password** and **Confirm Password** fields.
16. Clear the tick box **User Must Change Password at Next logon**.
17. Click on **Create** button. The user will be created and the **New User window appears**.
18. Click on **Close button**.
19. Close **Computer Management** utility.

Creating Password Policy

1. Select **Start > Programs > Administrative Tools > Local Security Policy**.
2. Expand **Account policies** and click on **Password Policy** folder. The **Password Policy Settings** appears in right hand windowpane.
3. Adjust the settings to include the following characteristics:
- Password history of 10
- Maximum password age of 28 days
- Minimum password length of 6
- Set password complexity settings on
4. Any settings can be adjusted by double clicking on it.
5. Close **Local Security Settings** utility after configuring the password settings.

Testing Password Policy

1. Log off from Administrator and Logon as Abdullah a password of "password".
2. Press **Ctrl+Alt+Del** to bring up **Security** window.
3. Select **Change Password** button. The **Password Change** window appears.
4. Enter "password" in **Old Password** field and "pass" in **New Password and Confirm** Password fields.
5. Click **OK**. A dialog box appears indicating the password is not acceptable and requesting you to enter a password that conforms to certain conditions.
6. Click **OK**. The **Change Password** window reappears.
7. Enter "password" in **Old Password** field and "pass234" in **New Password** and **Confirm Password** fields.
8. Click **OK**. The **Security** dialog box reappears.

9. Click **Cancel** to clear this.

Practical: Setting up Account Lockout Policy

Setup an account lockout policy and then test it.

Creating Account Lockout Policy

1. Logon as Administrator.
2. Select **Start > Programs > Administrative Tools > Local Security Policy**.
3. Expand **Account Policies** and click **Account Lockout Policy** folder. The **Account Lockout** policy settings should appear in the right hand windowpane.
4. Adjust the following settings to include the following characteristics.

- Account Lockout duration 0
- Account Lockout threshold 4

5. Close **Local Security settings utility after configuring the account lockout settings.**

Testing Account Lockout Policy

1. Log off from Administrator and logon as Abdullah account with a password of "password". Note that this is an incorrect password. A message will appear telling that the account name or password is incorrect.
2. Click **OK**.
3. Enter the same password for three times. A message will appear on third time telling that the account has been lockout.
4. Click **OK**.

Resetting Account Lockout Policy

1. Logon as Administrator.
2. Select **Start > Programs > Administrative Tools > Computer Management**.
3. Expand **Local Users and Groups** and click **User** folder.
4. Double click on **Abdullah** user account to open properties window.
5. Remove the tick next to **Account Is Locked Out** and click **OK**.
6. Close **Computer Management** utility.
7. Log off and logon again as Abdullah account using a password of "pass234" to check that the account is unlocked.

Practical: Modify User Rights

Change the user rights to limit the users who can logon to the system and the test whether the change works successfully.

Modify User Rights

1. Logon as Administrator.
2. Select **Start > Programs > Administrative Tools > Local Security policy**.
3. Expand **Local Policies** and click **User Rights Assignment** folder. The **User Rights Assignment** settings should appear in the right hand windowpane.
4. Scroll the list of rights to find **logon local right**.
5. Double click on **Logon Locally right**.
6. Remove the tick from **Local policy setting option for the Users group**. All new user accounts on the system will be made member of **User** group.
7. Click **Add** button and add Abdullah account.
8. Click OK. The **Abdullah** account should appear under **Logon Locally** right window.

9. Click **OK** and close **Local Security Settings** utility.

Testing User Rights

1. Log off as Administrator and logon as Usman with a password of "password". A message will appear telling that this account is not permitted to logon locally.
2. Click **OK**.
3. Logon using Abdullah account with a password of "pass234". This account is allowed to logon because it has been given the right to logon locally.

Practical: Securing Logon Process

There are number of techniques to make the logon process more secure.

Changing some Logon Setting

1. Logon as Administrator.
2. Select **Start > Programs > Administrative Tools > Local Security policy.**
3. Expand **Local policies** and click **Security Option Assignment** folder. The **Security Option Settings** should appear in the right hand windowpane.
4. Locate and change the following settings:

- Message Text For Users Attempting To Logon (Message Should read: Access restricted to authorized users only)
- Message Title For Users Attempting To Logon (Message should read: Security Warning)
- Do Not Display Last User Name in Logon Screen (Enable this setting)

5. Close **Local Security Setting** utility after changing the settings.

Test User right

1. Log off the system. An additional logon security screen will appear.
2. Click **OK**. The **Logon To Windows** dialog box appears. The previously logged on user Administrator does not appear. A User Name and password is required each time to logon to the system.

Practical: Setting up and Reviewing Security Log

Configure some security auditing settings, generate some events and then review them with the event view utility.

Setting up Auditing

1. Logon as Administrator.
2. Select **Start > Programs > Administrative Tools > Local Security policy.**
3. Expand **Local policies** and click **Audit policy** folder. The **Audit policy settings** should appear in the right hand windowpane.
4. Double click on **Audit Logon events** option to bring up the configuration screen.
5. Place a tick in both the **Success** and **Failure** box.
6. Click **OK**.
7. Close Local security utility.

Testing Auditing

1. Log off the system.
2. Logon as Administrator account but enter an invalid password.
3. Logon with the Administrator account with a valid password of password.
4. Select **Start > Programs > Administrative Tools > Event Viewer.**

5. Click on **Security log** in the left-hand windowpane to open its contents in the right hand windowpane. Two events should appear.
6. Double click on the event at the top of the list to get additional information on it.
7. Click on drop down arrow to get information on the other event.
8. Click **OK** to close the event information window and close **Event Viewer utility**.
9. Close **Local security setting utility**.

Resetting Abdullah Password

1. Select **Start > Programs > Administrative Tools > Computer Management.**
2. Expand **Local Users and Groups** and click **Users** folder.
3. Right click on **Abdullah** account and select **Set Password.**
4. Enter "password" in **New Password and Confirm Password** fields.
5. Click **OK**. The **Password Change confirmation** window appears.
6. Click **OK**.
7. Close **Computer Management** utility.

15.7 Profiles

The user accounts are used to store identification information of users in a network. A **user profile** is a collection of folders and data that stores user's current desktop environment and application settings as well as personal data. It also contains all network connections that are established when the user logs on to a computer such as **Start** menu items and mapped drives to network servers. The user profiles maintain consistency for users in desktop environments.

Windows 2000 creates a user file when a user logs on a computer for the first time. The user profile is also known as **local user profile**. When a user logs on to a client computer, the user always receives individual desktop settings and connections regardless of how many users share the same client computer.

Windows 2000 creates a default user profile for the user and stores it in the system partition **root\Documents and Setting\user_logon_name** folder. The **user_logon_name** is the name of the user that is used for logging on to the system.

15.7.1 Types of Profiles

Different types of profiles are as follows:

1. Local Profile

A **local profile** is machine specific. The information is stored locally on the workstation computers. The corresponding profile is loaded when a user logs on. A local profile can be used or modified when the user has logged on at that workstation computer.

2. Roaming Profile

A **roaming profile** is configured for the users who use more than one machine on the network. The profile information is stored on the server. The user with roaming file gets his personal settings when he logs in from any computer in the network. A roaming profile is especially helpful in domain environment.

3. Mandatory Profile

The **mandatory profile** is a read-only roaming profile. It cannot be modified by the user. It is used if the administrator does not want users to create their profiles. The administrator defines the environment for the users. The user can change the existing environment after logging on. The changes made by the user are not saved to the profile when the user logs off.

Creating Local Profile

The following procedure is used to create a local profile:

1. Log on as Administrator.
2. Create two new users "Ali" and "Amjad" using the procedure as discussed earlier.
3. Select Start > Programs > Accessories > Window Explore.
4. Expand **My Computer > C: > Documents and Settings**. Notice that Documents and Settings folder does not contain user profile folder for the new users.
5. Log off as Administrator and log on as Ali.
6. Right click on desktop and select **New > Shortcut**.
7. Type "Notepad" in **Create Shortcut** dialog box.
8. Accept Notepad as the name of shortcut and click **Finish**.
9. Log off as Ali and log on Amjad. Notice that user Amjad sees desktop configuration stored in the default user profile.
10. Log off as Amjad and log on as Ali. Notice that Ali sees the desktop configuration set up in step 6, 7 and 8.
11. Log off as Ali and log on Administrator.
12. Select Start > Programs > Accessories > Windows Explore.
13. Expand **My Computer > C: > Document and Settings**. Notice that the folder now contains user profile folders for Ali and Amjad.

15.7.2 Logon Script

A **logon script** is a file that is created and assigned to a user account to configure the working environment. A login script can be used to establish network connections or start applications. The assigned logon script is runs each time a user logs on.

15.7.3 Setting up Home Folder

The users normally store their personal files and information in a private folder called **home folder**. The user can specify the location of home folder as local folder or network folder in **Profile** tab of user proprieties dialog box.

1. Choose local path option and type the path in text box next to that option to specify a local path folder.
2. Open Computer Management window and expand Local Uses and Groups.
3. Open **Users** folder and double-click user1. The user1 properties dialog box appears.
4. Select **Profile** tab and click **Local Path radio button** to select it.
5. Specify home folder path by typing "C:\users\user1" in the text box for local path option and click **OK**.
6. Use Windows Explorer to verity that the folder has been created.

15.8 File and Folder Management

The file and folder management includes the following:

15.8.1 Organizing Files and Folders

The information stored on the computer can be searched easily if files and folders are organized properly. Planning is an important key to organization. For example, you can plan to store all applications on C: drive and all data on D: drive. You can also organize the data by function or by type. When the planning is implemented on the computer, it may require to rename, delete, move and copy files and folders etc. as follows:

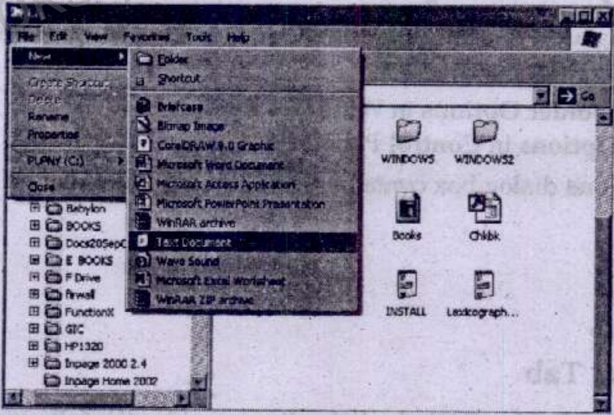
Creating Files and Folders

The folders can be created by using:

- Windows Explorer
- MD Command in DOS
- My Computer

The files can also be created in different ways. The most common way to create a file is by using an application software like WordPad or Notepad as follows:

1. Click **Start > Programs > Accessories > Windows Explorer**. OR Right click **Start** button and select **Explore**. The **Windows Explorer** will open.
2. Expand **My Computer** and select the drive and folder where the file is to be created.
3. Select **File > New > Text Document**.


File
Edit
View
Favorites
Tools
Help
New
Create Shortcut
Delete
Rename
Properties
PUPNY (C:)
Close
Folder
Shortcut
Briefcase
Bitmap Image
CorelDRAW 9.0 Graphic
Microsoft Word Document
Microsoft Access Application
Microsoft PowerPoint Presentation
WinRAR archive
Text Document
Wave Sound
Microsoft Excel Worksheet
WinRAR ZIP archive
Babylon
BOOKS
Docs20SepC
E BOOKS
F Drive
firwall
FunctionX
GIC
HP1320
Inpage 2000 2.4
Inpage Home 2002
Go
WINDOWS
WINDOWS2
Books
Chkbk
INSTALL
Lexicograph...

4. A new file icon appears in Windows Explorer windows. Type the name of new file.
5. Double click the file. The file will open in **Notepad**.

Renaming File and Folder

The following procedure is used to rename file or folder:

1. Right click the file or folder to be renamed. A popup menu will appear.
2. Select **Rename** from popup menu. The name of selected file or folder will be selected.
3. Type the new name and press **Enter**. The file or folder will be renamed.

Deleting File and Folder

The following procedure is used to delete file or folder:

1. Right click the file or folder to be deleted. A popup menu will appear.
2. Select **Delete** from popup menu. A dialog box will appear to confirm.
3. Click on **Yes** button. The file or folder will be deleted.

Copying File and Folder

The place from where the file or folder is copied is called **source**. The place where the file or folder is copied is called **destination**.

The following procedure is used to copy file or folder:

1. Right click the file or folder to be copied. A popup menu will appear.
2. Select **Copy** from popup menu.
3. Open the folder where the file is to be copied.
4. Right click on empty area. A popup menu will appear.
5. Click on **Paste** button. The file or folder will be copied.

Moving File and Folder

The following procedure is used to copy file or folder:

1. Right click the file or folder to be copied. A popup menu will appear.
2. Select **Cut** from popup menu.
3. Open the folder where the file is to be copied.
4. Right click on empty area. A popup menu will appear.
5. Click on **Paste** button. The file or folder will be moved.

15.8.2 Managing Folder Options

Folder Options is used to configure options such as Desktop view etc. It can be opened by using one of the following ways:

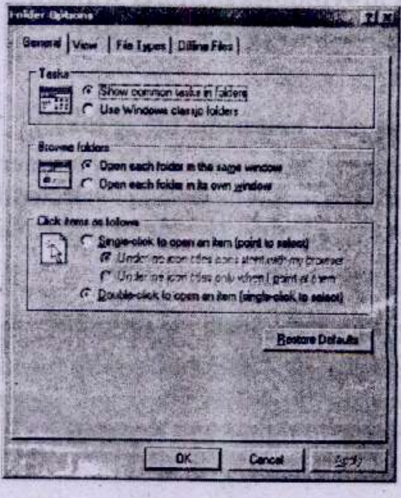
- Select **Tools > Folder Options in Windows Explorer.**
- Select **Folder Options in Control Panel.**

The **Folder Options** dialog box contains the following four tabs:

- General tab
- View tab
- File Types tab
- Offline Files tab

Options of General Tab

The **General** tab of **Folder Options** dialog box provides the following options:



Tasks
● Show common tasks in folders
○ Use Windows classic folders

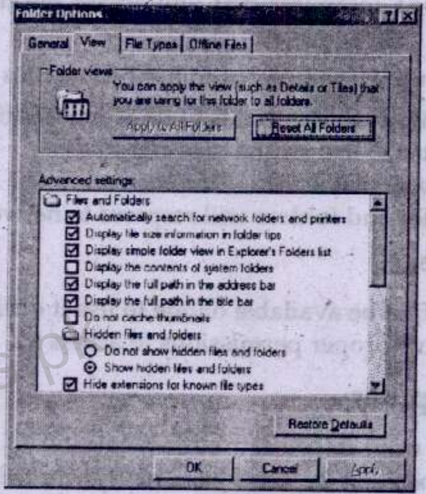
Browse folders
● Open each folder in the same window
○ Open each folder in its own window

Click items as follows
○ Single-click to open an item (point to select)
● Underline icon titles consistent with my browser
○ Underline icon titles only when I point at them
● Double-click to open an item (single-click to select)

- A choice to show common tasks in folders or use Windows classic folders
- The option to open all folders in same window while browsing or open each folder in a separate window.
- The option to open items with single mouse click or double click.

Options of View Tab

The **View** tab of **Folder Options** dialog box is used to configure what users see when they open files and folders. It provides the following options:


Folder Options
General
View
File Types
Offline Files
Folder views
Apply to All Folders
Reset All Folders
Restore Defaults
OK
Cancel
Apply

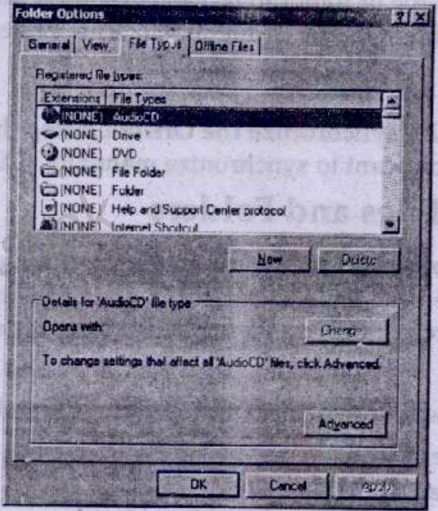
You can apply the view (such as Details or Tiles) that you are using for this folder to all folders.

Advanced settings:

- Files and Folders
  - ☑ Automatically search for network folders and printers
  - ☑ Display file size information in folder tips
  - ☑ Display simple folder view in Explorer's Folders list
  - ☐ Display the contents of system folders
  - ☑ Display the full path in the address bar
  - ☑ Display the full path in the title bar
  - ☐ Do not cache thumbnails
  - Hidden files and folders
    - ○ Do not show hidden files and folders
    - ● Show hidden files and folders
  - ☑ Hide extensions for known file types

Options of File Types Tab

The **File Types** tab is used to associate filename extension with application file types. When an extension is associated with a file type, users can double click a file to open it in its application. For example, **.txt** is associated with Notepad and user double clicks **myFile.txt**, Notepad will start and the file will be opened in it.


Folder Options
General
View
File Types
Offline Files
New
Delete
Change
Advanced
OK
Cancel
Apply

Registered file types:

Extensions	File Types
(NONE)	AudioCD
(NONE)	Drive
(NONE)	DVD
(NONE)	File Folder
(NONE)	Folder
(NONE)	Help and Support Center protocol
(NONE)	Internet Shortcut

Details for 'AudioCD' file type

Opens with:

To change settings that affect all 'AudioCD' files, click Advanced.

File Types tab is used to add, delete and change file-type associations. The new ilename extensions may also be added automatically when new application is installed.

Options of Offline Files Tab

The **Offline Files** tab is used to configure the computer to use offline files and folders. This feature allows the user to store network files and folders on clients. If the network location is not available, the users can still access network files.

At least two computers are required to configure offline files and folders:

- The network computer that contains the network version of the files and folders.
- The Windows XP client computer that will access the network files while they are online or offline.

The following tasks should be completed to use offline files and folders:

1. Attach to the shared file or folder that is to be accessed offline
2. Configure the computer to use offline files and folders
3. Make files and folders available for offline access
4. Specify how offline files and folders will respond to network disconnection.

1. Attaching to the Share

The file or folder must first be available online to use it offline. It must be shared on the server and the user should have proper permissions to access it.

2. Configuring the Computer

The Offline Files tab is used to configure the computer to use offline files and folders. The **Enable Offline Files** box must be checked. The **Synchronize All Offline Files before Logging Off** option must be checked to configure automatic synchronization between offline and online files. The **Fast User Switching** option should be disabled to use synchronization. The **Fast User Switching** option is available in **Control Panel** under **User Accounts**.

3. Making Files or Folders Available

The following steps are performed to make a file or folder available for offline access:

1. Right click the shared file or folder to be used offline. A popup menu will appear.
2. Select **Make Available Offline** from menu. The **Welcome to the Offline Files** wizard will start.
3. Click **Next**.
4. Uncheck **Automatically synchronize the Offline Files when I log on and log off my computer** option if you want to synchronize manually. Otherwise click **Next**.

15.8.3 Searching for Files and Folders

Windows 2000 offers powerful search capabilities to find a file or folder based on the filename or folder name. The user can also search by giving the text that is contained in file. Windows explorer has a search button on its toolbar. It displays search dialog box as shown when the user clicks it.

The user can specify the following options for various search:

- The filename or folder name
- The text to be searched
- The location to be searched

15.9 Shared Folders

Sharing is a process of allowing network users to access a folder. A network share provides a single location to manage shared data used by many users. Administrator can use sharing to install an application once instead of installing it locally at each computer. He can also manage the application from a single location.

Shared folders are used to allow network users to access files. If a folder is shared, users can connect to that folder over the network and access the file stored in it. The users must have permission to access a shared folder.

15.9.1 Creating a Shared Folder

A folder can be shared by logging as a member of administrators or power user group. The Sharing tab of the folder's properties box is used to enable and configure sharing.

The following options can be configured while sharing a folder:

Option	Description
Do Not Share This Folder	It specifies that the folder is only available via local access
Share this Folder	It specifies that the folder is available via local access and network access
Share Name	It specifies a descriptive name by which users will access the shared folder
Comments	It allows to enter more descriptive information about the shared folder
User Limit	It allows to specify the maximum number of connections to the shared folder at one time. The default is 10 users.
Permission	It allows to configure how users will access the folder over the network
Caching	It specifies how folders are cached when the folder is offline.

Practical: Creating a Shared Folder

The following procedure is used to create a shared folder:

1. Select **Start > Programs > Accessories > Window Explorer**.
2. Expand **My Computer** then expand Local Disk (D:).
3. Select **File > New > Folder**. A new folder will be created.
4. Specify its name as "Share Me".
5. Right-click the **Share Me** folder and select **Properties**. The **Properties** box will open.
6. Click **Sharing** tab.
7. Click **Share This Folder** radio button.
8. Type **Test Shared Folder** in **Shared Name** text box.
9. Type **This is a comment for a shared folder** in **Comment** text box.
10. Click **Allow** radio button under user limit and specify 5 users.
11. Click **OK** to close the dialog box.

15.9.2 Shared Folder Permissions

A shared folder can contain applications, data or a user's personal data called. Each type of data requires different shared folder permissions. Different types of permissions are as follows:

Setting	Description
Read	Display folder names, filenames, file data and attributes; run program files; and change folders within the shared folder
Change	Create folders, add files to folders, change data in files, append data to files, change file attribute, delete folder and files, plus, its allows the user to perform actions by the Read permission.
Full Control	Change file permissions, take ownership of files, and perform all tasks permitted by the change permission

Practical: Applying Share Permissions

1. Select **Start > Programs > Accessories > Window Explorer**.
2. Expand **My Computer** then expand Local Disk (D:).
3. Right-click **Share Me** folder and select **Sharing** tab.
4. Click **Permissions** button. The **Permission** dialog box will appear.
5. Highlight **Everyone** group and click **Remove** button.
6. Click **Add** button. The **Select Users, Computers and Group** dialog box will appear.
7. Select users **Usman** and **Abdullah**.
8. Click **Add** button and click **OK**.
9. Click user **Abdullah** and check **Allow** box for **Full Control** permission.
10. Click user **Usman** and check **Allow** box for **Read** permission.
11. Click **OK** to close the dialog box.

16

NETWORK PROTOCOLS, CONNECTIVITY & BACKUP

Chapter Overview

- 16.1 Network Protocols
- 16.2 Dial-up Networking and Internet Connectivity
  - 16.2.1 Creating Dialup Connection
- 16.3 Windows Backup
  - 16.3.1 Types of Backup
  - 16.3.2 Backing up Files and Folders

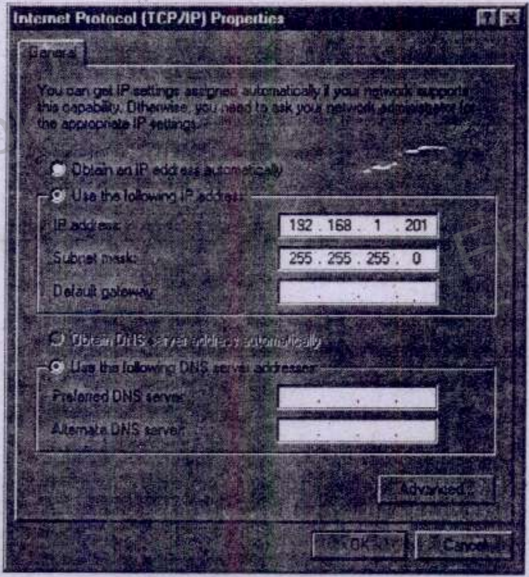
16.1 Network Protocols

A **protocol** is a set of rules and conventions for sending information over a network. Microsoft Windows 2000 uses TCP/IP for logon, file and print services, replication of information between one domain controller and another etc.

Configuring TCP/IP to Use Static IP Address

By default, the computers running Windows 2000, Windows 95 or Windows 98 obtain TCP/IP configuration automatically from **Dynamic Host Configuration Protocol (DHCP)**. The user user can still assign static IP address to selected network computers even in a DHCP-enabled environment. For example, the computer running DHCP service cannot be a DHCP client. It must have a static IP address. TCP/IP should be configured to use static IP address if DHCP Service is not available.

The following IP address, subnet mask and default gateway can be configured for each network adapter card that uses TCP/IP in a computer:



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:
IP address: 192 . 168 . 1 . 201
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: . . .
Alternate DNS server: . . .

Figure: Configuring a static TCP/IP address

The options used in configuring a static TCP/IP address are as follows:

IP Address

It is a logical 32-bit address that identifies a TCP/IP host. Each network adapter card in a computer running TCP/IP requires a unique IP address such as **192.168.1.108**. Each address has two parts which are Network ID and Host ID. The network ID identifies all hosts on the same physical network. The host ID identifies a host on the network. In this example, the network ID is 192.168.1 and host ID is 108.

Subnet Mask

Subnet mask is a network in a multiple-network environment that uses IP addresses derived from a single network ID. Subnets divide a large network into multiple physical networks connected with routers. A subnet mask blocks part of IP address so that TCP/IP can distinguish network ID from host ID. When TCP/IP hosts try to communicate, the subnet mask determines whether the destination host is on a local or remote network. Computers must have the same subnet mask to communicate on a network.

Default Gateway

Default gateway is the intermediate device on a local network. It stores network IDs of other networks in the enterprise or Internet. An IP address is configured for default gateway to communicate with a host on another network. TCP/IP sends packets for remote networks to the default gateway if no other route is configured. It forwards the packets to other gateways until the packet is delivered to a gateway connected to the specified destination.

Procedure

The following procedure is used to configure TCP/IP with static IP address:

1. Right-click **My Network Places** and click **Properties**. The **Network and Dial-up Connections** window will appear.
2. Right-click **Local Area Connection** and click **Properties**. The **Local Area Connection Properties** dialog box will appear.
3. Click **Internet Protocol (TCP/IP)** and verify that check box to its left is selected
4. Click **Properties**. The **Internet Protocol (TCP/IP) Properties** dialog box will appear.
5. Select **General** tab and click **Use the Following IP Address**.
6. Type TCP/IP configuration parameters and click **OK**.
7. Click **OK** to close **Local Area Connection Properties** dialog box.
8. Close **Network and Dial-Up Connections** window.

IP communications can fail if duplicate IP addresses exist on a network. It is important to obtain a valid static IP address.

Configuring TCP/IP to Obtain IP Address Automatically

The server running **DHCP Service** can automatically assign TCP/IP configuration information to DHCP client. The use of DHCP to configure TCP/IP automatically on client computers can simplify administration. It also ensures correct configuration information. Windows 2000 Professional does not include DHCP Service. It is available only in Windows 2000 Server.

Windows 2000 includes an Automatic Private IP Addressing feature. It provides DHCP clients with limited network functionality if a DHCP server is unavailable during startup.

DHCP Service can be used to provide clients with TCP/IP configuration information automatically. However, a computer must be configured as a DHCP client before it can interact with DHCP Service.

Procedure

The following procedure is used to configure a DHCP client:

1. Right-click **My Network Places** and click **Properties**. The **Network and Dial-up Connections** window will appear.
2. Right-click **Local Area Connection** and click **Properties**. The **Local Area Connection Properties** dialog box will appear.
3. Click **Internet Protocol (TCP/IP)** and verify that check box to its left is selected

4. Select **General** tab in **Internet Protocol (TCP/IP) Properties** dialog box and click **Obtain An IP Address Automatically**.
5. Click **OK** to close **Local Area Connection Properties** dialog box.
6. Close **Network And Dial-Up Connections** window.

Testing TCP/IP Configuration

Different utilities are used to test TCP/IP configuration:

ipconfig Utility

A small utility **ipconfig** is used to test the configuration and connections to other TCP/IP hosts and networks. It can be used after configuring TCP/IP and restarting the computer. It helps the user to ensure that TCP/IP is functioning properly.

The ipconfig can be used to verify TCP/IP configuration parameters on a host. It also helps to determine whether configuration is initialized or a duplicate IP address exists.

Ping Utility

The ping utility is used to test connectivity. It is a diagnostic tool to test TCP/IP configurations and diagnose connection failures. It can be used to determine whether a particular TCP/IP host is available and functional. Its syntax is as follows:

```
ping IP_address
```

Practical

Installing and Configuring TCP/IP

The practical includes the following:

- Use two TCP/IP utilities to verify computer's configuration.
- Configure the computer to use a static IP address and verify computer's new configuration.
- Configure the computer to use a DHCP server to automatically assign an IP address to the computer whether or not a DHCP server is available on the network.

Procedure

The practical requires the following:

- TCP/IP as the only installed protocol.
- Optional: A server running DHCP Service to provide IP addresses. If the computer is not part of a network and a server is not running DHCP service, certain procedures will not be performed.

Verifying TCP/IP Configuration

1. Open **Command Prompt** window.
2. Type **ipconfig /all | more** and press **Enter**.
3. Windows 2000 IP Configuration utility displays TCP/IP configuration of the physical and logical adapters configured on the computer.
4. Press **Spacebar** as necessary to display Local Area Connection.
5. Press **Spacebar** to display additional information and to return to command prompt.
6. Type **ping 127.0.0.1** to verify that TCP/IP is working and configured for adapter.

7. **Press Enter.**
   A response similar to the following indicates a successful ping:

```
Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128

Ping statistics for 127.0.0.1:
  Packets: Sent = 4, Received = 4, Lost = 0 <0% loss>,
Approximate round trip times in milliseconds:

  Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

8. Minimize the Command Prompt window.

Configuring TCP/IP to Use Static IP Address

1. Right click **My Network Places** and click **Properties**. The **Network And Dial-Up Connections** window appears.
2. Right click **Local Area Connection** and click **Properties**. The **Local Area Connection Properties** dialog box appears. It displays the network adapter and the network components used in the connection.
3. Click **Internet Protocol (TCP/IP)** and verify that check box to the left of the entry is selected.
4. Click **Properties**. The **Internet Protocol (TCP/IP) Properties** dialog box appears.
5. Click **Use The Following IP Address.**
6. Enter the following values for IP address, subnet mask and default gateway:

Variable value	Suggested value
Static IP address	192.168.1.201
Subnet mask	255.255.255.0
Default gateway (if required)	None

7. Click **OK**. The **Local Area Connection Properties** dialog box will display.
8. Click **OK** to close **Local Area Connection Properties** dialog box.
9. Minimize **Network And Dial-Up Connections** window.

Test Static TCP/IP Configuration

1. Restore Command Prompt.
2. Type **ipconfig /all | more** and press **Enter**. Windows 2000 **IP Configuration** utility displays physical and logical adapters configured on the computer.
3. Press **Spacebar** as needed to scroll through the configuration information and locate the **local area connection** information.
4. Record the current TCP/IP configuration settings for your local area connection in the following table.

Setting	Value
IP address	
Subnet mask	
Default gateway	

5. Press **Spacebar** as necessary to scroll through configuration information and return to command prompt.

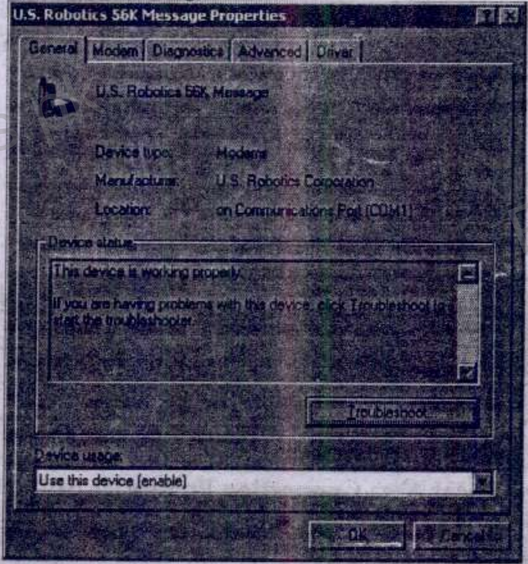
6. Type ping **127.0.0.1** and then press Enter to verify that IP address is working and configured for the adapter.
7. Type **ping ip_address** if a computer is available to test connectivity. Here, ip_address is the IP address of the computer being tested for connectivity.

16.2 Dial-up Networking and Internet Connectivity

Dialup networking is used to dial into the network or the Internet. A computer must be configured for dialup before using dialup networking. It requires to install and configure a modem properly.

The **device manage** is used to configure and manage the modems. The device manager can be accessed as follows:

- Right click **My Computer** and select **Manage** from pop-up menu.
- Select **System Tools > Device Manager.**
- Select **Modems** in device manager window and double-click the modem to be managed to display the modem properties dialog box. The dialog box has six tabs. These are General, Modem, Diagnostics, Advanced, Driver and Resources.



U.S. Robotics 56K Message

Device type: Modems
Manufacturer: U.S. Robotics Corporation
Location: on Communications Port (COM1)

Device status
This device is working properly.
If you are having problems with this device, click Troubleshoot to start the troubleshooter.

Device usage:
Use this device (enable)

Configuring General Modem Properties

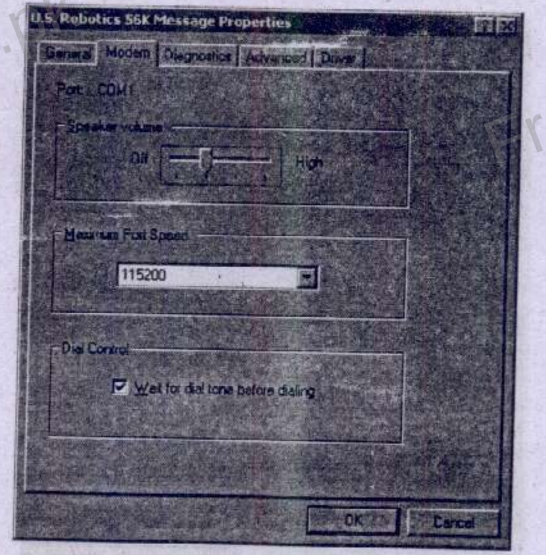
The **General** tab of modem properties dialog box displays device type, manufacture and location. It also shows the device status.

The **Troubleshooter** button is used to start a troubleshooting wizard if the modem is not working properly. The troubleshooter button helps to determine the cause of problem.

Configuring Modem Properties

The **Modem** tab shows the following options:

1. The speaker volume of the modem.
2. The maximum port speed (Specified in bits per seconds)
3. Dial control to wait for dial tone before dialing.


U.S. Robotics 56K Message Properties
General
Modem
Diagnostics
Advanced
Driver
Speaker volume
Off
High
Maximum Port Speed
Dial Control
OK
Cancel

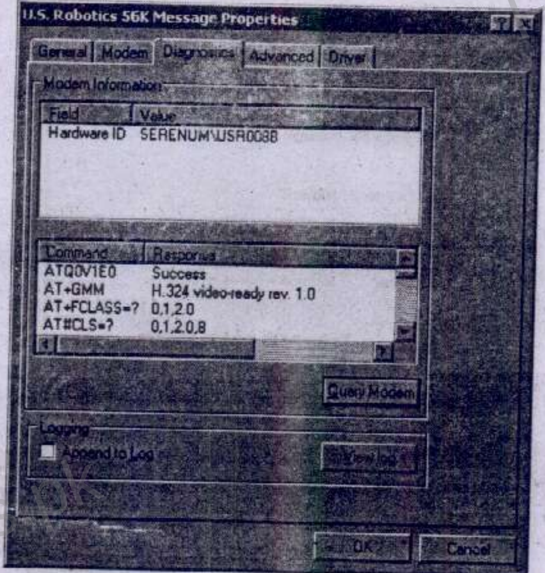
Port: COM1

Maximum Port Speed: 115200

☑ Wait for dial tone before dialing

Running Modem Diagnostics

The **Diagnostic** is used to query the modem. It is used in troubleshooting themodem. It ensures that the modem is properly responding to request. The **Query Modem button tests** the modem by issuing a series of modem commands.


U.S. Robotics 56K Message Properties
General
Modem
Diagnostics
Advanced
Driver
Modem Information
Query Modem
Logging
View log
OK
Cancel

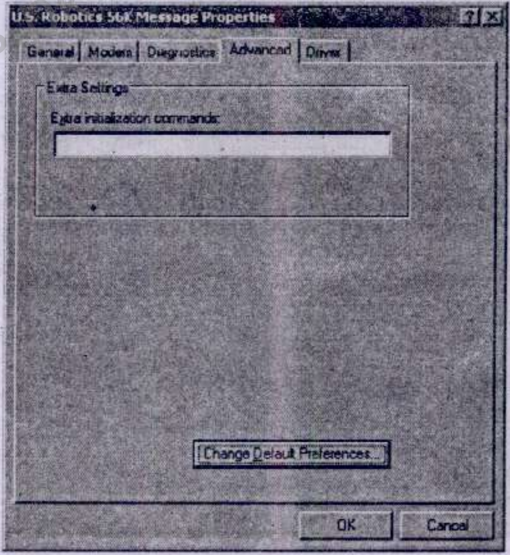
Field	Value
Hardware ID	SERENUM\USR0088

Command	Response
ATQ0V1E0	Success
AT+GMM	H.324 video-ready rev. 1.0
AT+FCLASS=?	0,1,2.0
AT#CLS=?	0,1,2.0,8

☐ Append to Log

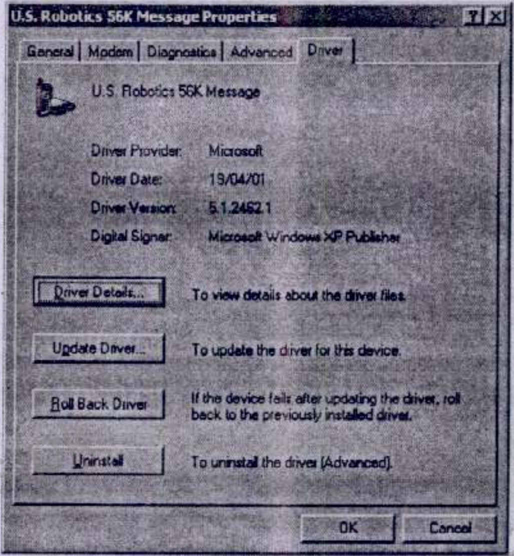
Configuring Advanced Modem Properties

The **Advanced** tab is used to specify additional initialization commands. It can also be used to configure advanced port setting and change default preferences.


U.S. Robotics 56K Message Properties
General
Modem
Diagnostics
Advanced
Driver
Extra Settings
Extra initialization commands:
Change Default Preferences...
OK
Cancel

Viewing Driver Details and Updating Drivers

The **Driver** tab is used to display information about the current driver of modem. The **Driver Details** button shows a dialog box with additional information about the driver. The Uninstall Driver is used to uninstall the driver.


U.S. Robotics 56K Message Properties
General
Modem
Diagnostics
Advanced
Driver

U.S. Robotics 56K Message

Driver Provider: Microsoft
Driver Date: 19/04/01
Driver Version: 5.1.2462.1
Digital Signer: Microsoft Windows XP Publisher

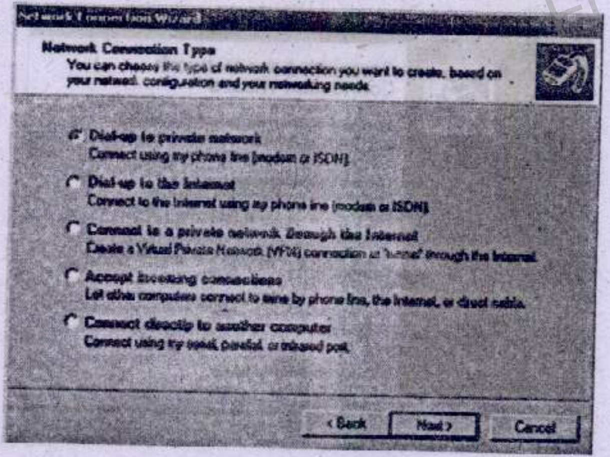
Driver Details... To view details about the driver files.
Update Driver... To update the driver for this device.
Roll Back Driver If the device fails after updating the driver, roll back to the previously installed driver.
Uninstall To uninstall the driver (Advanced).

16.2.1 Creating Dialup Connection

The following procedure is used to create a dialup connection:

1. Select **Start > Settings > Control Panel.**
2. Double click **Network and Dialup** connection icon. The **Network and Dial-up Connection** window will appear.
3. Double click **Make New Connection** icon. The **Welcome to the Network Connection wizard will start.**

4. Click **Next**. The **Network Connection Type** dialog box will appear.
5. Select **Dialup to private Network** option and click **Next**. The **Phone Number to Dial** dialog box will appear.

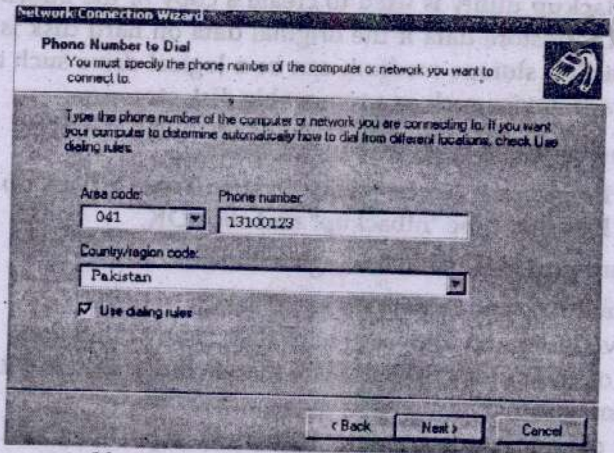


Network Connection Wizard

Network Connection Type
You can choose the type of network connection you want to create, based on your network configuration and your networking needs.

● **Dial-up to private network**
Connect using my phone line (modem or ISDN).
○ **Dial-up to the Internet**
Connect to the Internet using my phone line (modem or ISDN).
○ **Connect to a private network through the Internet**
Create a Virtual Private Network (VPN) connection or 'tunnel' through the Internet.
○ **Accept incoming connections**
Let other computers connect to mine by phone line, the Internet, or direct cable.
○ **Connect directly to another computer**
Connect using my serial, parallel, or infrared port.

6. Enter a telephone number in **Phone number** box. If you want to enter an area code, first check **Use dialing rules** checkbox.
7. Click **Next**.



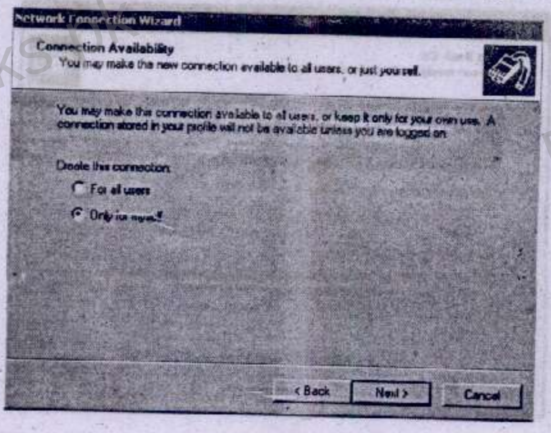
Network Connection Wizard

Phone Number to Dial
You must specify the phone number of the computer or network you want to connect to.

Type the phone number of the computer or network you are connecting to. If you want your computer to determine automatically how to dial from different locations, check Use dialing rules.

Area code: 041
Phone number: 13100123
Country/region code: Pakistan
☑ Use dialing rules

8. Select **Only for myself** option in **Connection Availability** dialog box and click **Next**.



Network Connection Wizard

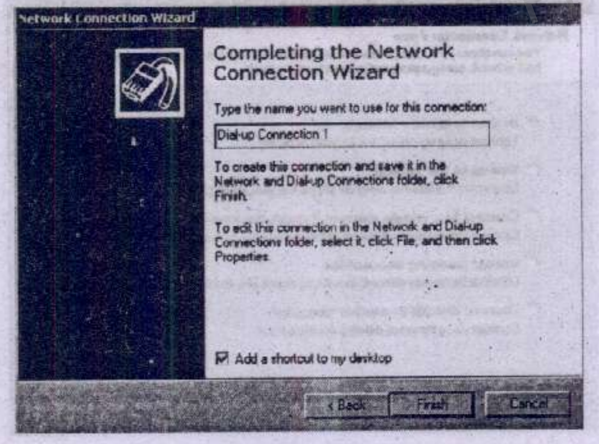
Connection Availability
You may make the new connection available to all users, or just yourself.

You may make this connection available to all users, or keep it only for your own use. A connection stored in your profile will not be available unless you are logged on.

Create this connection:
○ For all users
● Only for myself

Freebooks.pk Freebooks.pk

9. Specify the name for the connection in **Completing the Network Connection Wizard** dialog box.
10. Click **Finish**.



Completing the Network Connection Wizard

Type the name you want to use for this connection:

Dial-up Connection 1

To create this connection and save it in the Network and Dial-up Connections folder, click Finish.

To edit this connection in the Network and Dial-up Connections folder, select it, click File, and then click Properties.

☑ Add a shortcut to my desktop

16.3 Windows Backup

Freebooks.pk Freebooks.pk

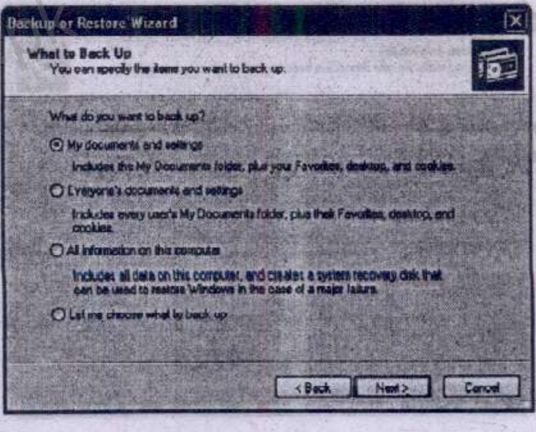
The **Windows Backup** utility is used to create a copy of the information on hard disk. This copy can be used to restore data if the original data on hard disk is accidentally erased or overwritten. The backup storage medium can be a logical drive such as hard drive. It can also be a separate storage device such as a removable disk etc.

The following procedure is used to launch Backup:

1. Select **Start > Programs > Accessories > System Tools > Back up**. OR
2. Select **Start > Run** and type "ntbackup" and click **OK**.

Windows Backup provides the following facilities:

- Archive selected files and folders on hard disk.
- Restore archived files and folders to hard disk or any other disk.
- Make a copy of computer's System State that includes system files, registry and other system components.
- Make a copy of computer's system partition, boot partition and files needed to start the system in case of computer or network failure.



What to Back Up

You can specify the items you want to back up.

What do you want to back up?

● My documents and settings

Includes the My Documents folder, plus your Favorites, desktop, and cookies.

○ Everyone's documents and settings

Includes every user's My Documents folder, plus their Favorites, desktop, and cookies.

○ All information on this computer

Includes all data on this computer, and creates a system recovery disk that can be used to restore Windows in the case of a major failure.

○ Let me choose what to back up

Freebooks.pk Freebooks.pk

16.3.1 Types of Backup

The Windows Backup supports five methods to back up data on computer or network.

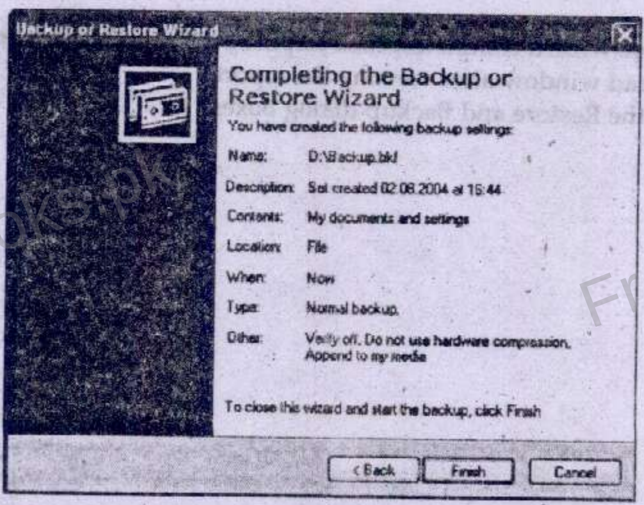
- **Copy Backup:** A **copy backup** copies all selected files but does not mark each file as backed up. It is useful if the user wants to back up files between normal and incremental backups because copying does not affect other backup operations.
- **Daily Backup:** A **daily backup** copies all selected files that have been modified in the whole day when the backup is performed. The backed-up files are not marked as backed up.
- **Differential Backup:** A **differential backup** copies files created or changed since the last normal or incremental backup. It does not mark files as backed up.
- **Incremental Backup:** An **incremental backup** backs up only those files created or changed since the last normal or incremental backup. It marks files as backed up.
- **Normal Backup:** A **normal backup** copies all selected files and marks each file as backed up.

16.3.2 Backing up Files and Folders

A file name and location is designated for the backup file. Windows Backup files usually have the extension **.bkf** but the user can change it. A backup file can be saved to a hard disk, a floppy disk or any other removable or non-removable media.

The following four steps describe a simple backup operation:

- **Select files, folders and drives for backup:** Windows Backup provides a tree view of drives, files and folders on computer. The user can select files and folders to back up.
- **Select storage media or file location for backed-up data:** Windows Backup provides two options for selecting storage media. The user can back up data to a file on a storage device such as hard disk, Zip disk etc. The user can also back up data to a tape device if a tape device is installed on the computer.
- **Set backup options:** Windows Backup provides an **Options** dialog box. It is used to customize backup operations. The user can select the type of backup, select whether a log file should record backup actions, designate file types to exclude from backup.
- **Start the backup:** Windows Backup starts backing up files and folders in this step.



Completing the Backup or Restore Wizard

You have created the following backup settings:

Name: D:\Backup.bkf
Description: Set created 02.08.2004 at 15:44
Contents: My documents and settings
Location: File
When: Now
Type: Normal backup.
Other: Verify off. Do not use hardware compression. Append to my media

To close this wizard and start the backup, click Finish

You must be an administrator or a backup operator to back up all files and folders. If you are a member of Users or Power Users group, you must be the owner of the files and folders to back up or you must have proper permissions.

Practice: Backup Files

Create a folder on D: drive called **DATA**. Create some small text files in this folder. The size of all of the files combined should not exceed IMB.

1. Select **Start > Programs > Accessories > System Tools > Backup.**
2. Click **Backup wizard** button. The **Welcome to the Windows 2000 Backup** and **Recovery Tools** dialog box will appear.
3. Click **Next.** The **What to Backup** dialog box will appear.
4. Click **Back Up Selected files, Drives, or Network Data radio button** and click **Next.**
5. Select **My Computer > D: > DATA** folder in the items to back up dialog box.
6. Click **Next.** The **Where to Store the Backup** dialog box will appear.
7. Click **Browse.**
8. Select the drive and give backup a file name.
9. Click Next in **Where to Store the Backup** dialog box.
10. Verify selections in **Completing the Backup wizard** dialog box and click **Finish.**
11. When Backup wizard completes, click **Report** button in **Backup progress** dialog box. It will show the backup log in a notepad window.
12. Close the window after viewing report.
13. Close all of Backup wizard dialog boxes.

Practice: Restore Files

1. Select **Start > Programs > Accessories > System Tools > Backup.**
2. Click **Restore Wizard** button. The **Welcome to the Restore wizard** dialog box will appear.
3. Click **Next.**
4. Click the backup filename and click **Next.**
5. Verify that every ting is configured properly in **Completing the Restore wizard** dialog box and click **Finish.**
6. Verify that filename for backup is correct in **Enter Backup File Name** dialog box and click **OK.**
7. When Restore wizard completes, click **Report button in Restore progress** dialog box.
8. Close Notepad window after viewing the report.
9. Close all of the Restore and Backup dialog boxes.

Windows 2000 Professional: Review Questions

Q.1. Which of the following file systems are supported by Windows 2000 Professional and under what circumstances are you likely to use each of them?

FAT and FAT32 are supported but are only likely to be used in a dual-boot situation with another operating system. NTFS is a better choice when only Windows 2000 is installed.

Q.2. Which two methods can you use to invoke the setup installation program and under what circumstances would you use each of them?

The setup installation program can be invoked by booting the system directly from the Windows 2000 CD or by booting from four-setup floppy diskettes. The floppy diskettes are used if the system will not auto boot from CD-ROM drive.

Q.3. Which steps are involved during text-based and graphics-based phase of installation?

Selecting a hard disk partition & file system and accepting the licensing agreements occur during the text-based phase of installation. Entering an organization & owner and selecting the network components and entering a produce code occur during the graphics-based phase of installation.

Q.4. What network settings are configured automatically when the user selects 'Typical settings' during installation?

TCP /IP is installed and automatic configuration is selected. The 'client for Microsoft Networks' and 'File and print sharing for Microsoft Networks' services are installed.

Q.5. What can you do if you have hardware that is not supported by device drivers that exists on the Windows 2000 CD?

You can still get device drivers that have written and tested after Windows 2000 was launched. These can be obtained from manufactures and used during the installation. They can be copied on to diskette and loaded at various points during the installation process.

Q.6. What can you do if the system has CD-ROM drive attached that is not supported by Window 2000?

Either replace CD-ROM drive with a supported one or boot the system from an operating system that recognizes CD-ROM drive. Copy the source files to the local hard disk and run a program called WINNT from here to start the installation process.

Q.7. Which files are held on the system partition and which are held on boot partition?

Boot.ini, Ntdetect.com and Ntldr are held on system partition. Ntoskrnl.exe and Hal.dll are held on boot partitions.

Q.8. What is the difference between logging on with a local account and logging on with domain account?

When you logon with a local account it is verified by checking an accounts database held on that local system. When you logon with a domain account it is sent from the local system to a domain controller to be verified.

Q.9. Why is shutting down a Windows 2000 system important?

Shutting down a system ensures that all applications and backgrounds tasks are stopped properly. It prevents the loss of data or information.

Q.10. What is the main advantage of domain accounts over local accounts?

Domain accounts can be shared by all systems that exist within domain. One account can potentially be used to logon to all systems. Local accounts exists on a specific system, if you want to logon to many systems you need an account created on each system.

Q.11. What are common characteristics that all user accounts share?

All user accounts will have a logon name and password. They will also have other persona details stored, the amount and complexity of these varies from system to system.

Q.12. What are the characteristics of local user accounts?

Local user accounts can only be created on Windows 2000 Professional systems and Windows 2000 Servers that are not domain controllers. They are not shared in anyway and can only be used on the system where they are created.

Q.13. What are the characteristics of domain user accounts?

Domain user accounts can only be created on domain controllers. These controllers support what is called a multiple master model, so accounts can be created on any of them and they keep each other informed of updates. Domain accounts can be used by any systems that are part of the domain.

Q.14. What are the built-in users accounts provided on every Window 2000 system and what are their key characteristics?

The built-in accounts are Administrator and Guest. The administrator account has no restrictions and can do absolutely everything to a system. The Guest account is a courtesy account provided to give someone casual access to a system without creating a specific account. It is normally used for network access. By default this account is disable.

Q.15. What tools do you used to create local user accounts and domain user accounts?

You use computer Management to create local user accounts and Active Directory user and computer to create domain user accounts.

Q.16. List different types of profiles and describe their characteristics?

- **Local Profile:** It is stored on local hard disk of the system on which the user has logged on.
- **Default Profile:** It is provided by Windows 2000 that is applied to any new user that logs on who does not currently have their own profiles.
- **Roaming Profile:** It is stored on central server and copied down onto each system that the user logs on at.
- **Mandatory Profile:** It is fixed and cannot be changed.

Q.17. What is the purpose of logon script?

It is used to carry out a number of tasks automatically when a user logs on.

Q.18. What are some of the benefits of using groups?

They combine a number of users together to make administration easier. Anything the group can do so can any member of the group. They also reduce the overhead on the operating system by giving it fewer entries to search through when checking security.

Q.19. What are the two main things that groups are used for?

Groups can be used to allow users to access system resources, such as printer, and to perform system tasks such as installing support for a new item of hardware.

Q.20. Write the names of built-in local groups on Windows 2000 Professional system?

Administrators, Backup operators, everyone group and create owner group are built-in system groups.

Q.21. What are the main advantages of NTFS security?

It protects folders and files and prevents unauthorized access and accidental damage. It utilizes this security capability both locally when working interactively and across network when accessing a system remotely. It is also possible to audit access to folders and files and track their use.

Q.22. Which of the following permissions are really only applicable to files?

Basic attributes include Read-only and Hidden. Generally, these are only used on files, although they re occasionally applied to folders. Extended attributes include author name, title and subject. These are applied by individual applications and are only relevant to files.

Q.23. What is difference between Read and Execute and List folder contents when applied to folder?

List folder contents allows the user to look at the contents of a folder. However, the user cannot read the contents of any file. It contains or executes any program it contains. Read and execute allows the user to look at the contents of the folder and ready any files and execute any program it contains.

Q.24. What is the difference between Delete and Delete subfolders and File permission?

Delete allows the user to delete a folder if he has specific permissions to delete everything in that folder. The Delete subfolders and files allows the user to delete a folder even if it contains items that he does not have the specific permission to delete.

Q.25. Describe some of the things to consider when applying NTFS permission?

Groups similar files together like all applications in one area and all data files in another area. Assign the most restrictive permissions first and ease these later if required. Give administrators full control typically and restrict users to the specific permission required.

Q.26. Name two methods to share out a folder?

Using windows explorer, right click on a folder and selecting sharing from pop-up menu.

Go to a command prompt and use the NET SHARE command.

Q.27. Which local ports are typically expected to be on every system when you are installing printers?

You would expect to see parallel port (LPT), serial ports (COM) and option to send output to a file.

Q.28. When you share out a printer what is important about share name and location filed?

The share name should be logical and short. It may need to be no longer then 8.3 if you have some older DOS clients. The location field can provide information telling users where to find the printer.

Q.29. Under what circumstances would you consider printer priorities?

When two groups of users share the same print device and you want to allow one group of users to print their document before the others.

Q.30. What are some of the main advantages of plug and play hardware?

There is no need to manually configure the hardware. It is all done with software. The operating system automatically detects the hardware and installs the appropriate driver. Any conflicts over resources should be resolved without the user intervention.

Q.31. What are the main facilities provided by accessibility support under Window 2000?

You can configure settings to make the user of the hardware easier and run a number of programs to help a user with disabilities.

Q.32. Why is it important to enter a password for the administrator account and what should you also do to the account after installation?

The administrator account is all-powerful and has no restrictions. It is important to protect it with a password so that only authorized people can use it. It is also a good idea to rename the account after installation.